

Column Name	Null?	Type	Column Comment
twgbwrul_java_classpath		VARCHAR2(255)	Obsolete.
twgbwrul_disp_admin_desc_ind	Not null	VARCHAR2(1)	Indicates if the descriptive name of Web Tailor objects are displayed on the Web Tailor administrative Web pages.
twgbwrul_asrc_code		VARCHAR2(4)	Address source code information.
twgbwrul_disp_dead_ind	Not null	VARCHAR2(1)	Indicate if information for a deceased person should appear on the Web.
twgbwrul_cache_ind	Not null	VARCHAR2(1)	Indicates whether Web page caching is enabled for the entire system
twgbwrul_http_redirect_ind	Not null	VARCHAR2(1)	A value of Y redirects the user to owa_util.redirect_url.N uses an HTML META refresh tag instead.
twgbwrul_start_page		VARCHAR2(120)	Indicates which page users will see directly after login.
twgbwrul_start_page_menu_ind		VARCHAR2(1)	Indicates whether the start page is a Web Tailor-generated menu.
twgbwrul_date_input_fmt		VARCHAR2(30)	Stores the date input format mask to be used when entering free-form dates on the Web.
twgbwrul_dtd_type		VARCHAR2(255)	High-level information about the Document Type Definition (DTD) that the Web page was coded to.
twgbwrul_dtd_fpi		VARCHAR2(255)	The Formal Public Identifier (FPI) portion of the DTD.

Column Name	Null?	Type	Column Comment
twgbwrul_dtd_url		VARCHAR2(255)	The URL portion of the DTD. Together, the previous two columns and this one make up the !DOCTYPE tag used by the Self-Service applications.
twgbwrul_language		VARCHAR2(5)	The language used on the Web pages. Values should conform to ISO 639 standards.
TWGBWRUL_CGIBIN_DIR_ADM	Not Null	VARCHAR2(255)	Administrative path suffix of the Database Access Descriptor (DAD) for Oracle FGAC. The suffix determines which DAD is used for VBS. For example, if the normal DAD you use is <code>dadb70</code> , and the VBS DAD is <code>dadb70o</code> , this field will contain <code>o</code> .

Web Tailor Web SessionID Table (TWGBWSES)

This table contains session-specific information, including the following.

- Number of logon attempts
- Time and date of last access
- Generated session ID

Column Name	Null?	Type	Column Comment
twgbwses_pidm	Not null	NUMBER(8)	PIDM of user.
twgbwses_webid		VARCHAR2(6)	Randomly generated Web session id.
twgbwses_last_access		DATE	Last date the user logged into the system.
twgbwses_login_attempts		NUMBER(2)	Counter for number of times an unsuccessful login has been attempted for same ID.
twgbwses_login_req_ind		VARCHAR2(1)	For systems which include insecure pages - Y if user should be required to login at next secure page.
twgbwses_sessionid		NUMBER	Current Oracle session ID for this database transaction.
twgbwses_prev_webid		VARCHAR2(6)	Previous Web session ID.

Web Tailor SSB Page Access Audit Table (TWGRACES)

This table stores information related to page access in SSB by users of Banner.

Column Name	Null?	Type	Column Comment
TWGRACES_AUDIT_TIME	NOT NULL	TIMESTAMP(9)	The time the audit row was written.

Column Name	Null?	Type	Column Comment
TWGRACES_SSB_LOGON_ID	NOT NULL	VARCHAR2(256)	The user id used to login to SSB. This could also be an e-mail address or other form of logon identification.
TWGRACES_SSB_PAGE	NOT NULL	VARCHAR2(256)	The page the user is trying to access.
TWGRACES_IPADDR		VARCHAR2(15)	The IP address of the user logging on from SYS_CONTEXT('USERENV', 'IP_ADDRESS', 15).
TWGRACES_ACTIVITY_DATE	NOT NULL	DATE	Activity date that the user created or updated the row.
TWGRACES_USER_ID	NOT NULL	VARCHAR2(30)	User id that created or updated the row.
TWGRACES_SURROGATE_ID		NUMBER(19)	Surrogate id for the created row.
TWGRACES_VERSION		NUMBER(19)	Incremented number used for optimistic locking.
TWGRACES_DATA_ORIGIN		VARCHAR2(30)	Data origin created or updated the row.
TWGRACES_VPDI_CODE		VARCHAR2(6)	VPDI code for the created or updated row.

Web Tailor SSB Login Audit Table (TWGRALGN)

This table stores information related to logins to SSB by users of Banner.

Column Name	Null?	Type	Column Comment
TWGRALGN_AUDIT_TIME	NOT NULL	TIMESTAMP(9)	The time the audit row was written.
TWGRALGN_SSB_LOGON_ID	NOT NULL	VARCHAR2(256)	The user id used to login to SSB. This could also be an e-mail address or other form of logon identification.
TWGRALGN_SSB_PAGE		VARCHAR2(256)	The page the user is trying to access.
TWGRALGN_HOST		VARCHAR2(100)	The name of the local host from UTL_INADDR.GET_HOST_NAME
TWGRALGN_SADDR		RAW(8)	Session address from v \$session.saddr.
TWGRALGN_SID		NUMBER	Session identifier from v \$session.sid.
TWGRALGN_SERIAL		NUMBER	Session serial number from v \$session.serial.
TWGRALGN_AUDSID		NUMBER	Auditing session ID from v \$session.audsid.
TWGRALGN_PROCESS		VARCHAR2(24)	Operating system client process ID from v\$session.process.
TWGRALGN_MACHINE		VARCHAR2(64)	Operating system machine name from v\$session.machine.

Column Name	Null?	Type	Column Comment
TWGRALGN_LOGON_TIME		DATE	Time of logon from v \$session.logon_time.
TWGRALGN_IPADDR		VARCHAR2(15)	The IP address of the user logging on from SYS_CONTEXT("USERENV", "IP_ADDRESS", 15).
TWGRALGN_OSUSER		VARCHAR2(30)	Operating system client user name from v\$session.mouser.
TWGRALGN_TERMINAL		VARCHAR2(30)	Operating system terminal name from v\$session.terminal.
TWGRALGN_PROGRAM		VARCHAR2(64)	Operating system program name executing during login from v \$session.program.
TWGRALGN_MODULE		VARCHAR2(48)	Name of the currently executing module as set by calling the DBMS_APPLICATION_INFO.SET_MODULE procedure during logon from v \$session.module.
TWGRALGN_SERVICE_NAME		VARCHAR2(64)	Service name of the session from v\$session.service_name.
TWGRALGN_LOGON_COMMENT		VARCHAR2(256)	A comment recorded at logon time. Enables tracking of reason that login failed. The current values are: API Error, LDAP Error, LDAP Mapping Error, LDAP User Not Found, LDAP User Not Unique, Unknown Error, User Dead, User Disabled, User

Column Name	Null?	Type	Column Comment
			Error ⚠ i.e. bad pin, user, User Prohibited, Valid Login.
TWGRALGN_ACTIVITY_DATE	NOT NULL	DATE	Activity date that the user created or updated the row.
TWGRALGN_USER_ID	NOT NULL	VARCHAR2(30)	User id that created or updated the row.
TWGRALGN_SURROGATE_ID		NUMBER(19)	Surrogate id for the created row.
TWGRALGN_VERSION		NUMBER(19)	Incremented number used for optimistic locking.
WGRALGN_DATA_ORIGIN		VARCHAR2(30)	Data origin created or updated the row.
TWGRALGN_VPDI_CODE		VARCHAR2(6)	VPDI code for the created or updated row.

Channel SourceInfo Master Table (TWGRCHNL)

This is the main channel source table. It stores all the attributes related to the channel source.

SourceInfo is a generic keyword that tells Banner what XML content is being requested by the channel.

The SourceInfo name is a parameter defined in the channels. It identifies channels in channel database tables.

Note: This table was delivered without any data. A script to load the table was delivered with Luminis Channels for Banner..

Column Name	Null?	Type	Column Comment
TWGRCHNL_NAME	Not null	VARCHAR2(50)	The name of the SourceInfo as defined during the channel

Column Name	Null?	Type	Column Comment
			publishing process. For example, <code>gi_mybanner</code> is the SourceInfo for the My Banner channel.
TWGRCHNL_MODULE_CODE	Not null	VARCHAR2(3)	The functional module to which the channel belongs.
TWGRCHNL_ACTIVITY_DATE	Not null	DATE	The date the record was created or last updated.
TWGRCHNL_USER_ID	Not null	VARCHAR2(30)	The ID of the person who created or last updated the record.
TWGRCHNL_SECURE_IND	Not null	VARCHAR2(1)	Specifies if the channel must be secured. Valid values are: <code>Selected</code> - the Banner Main package, TWBCMAIN, will validated the external system ID. <code>Cleared</code> - no authentications will be performed for the external system ID via TWBCMAIN.
TWGRCHNL_TYPE		VARCHAR2(10)	Specifies if the channel is informational or navigational. Valid values are: <code>I</code> - Informational <code>N</code> - Navigational

Channel Mapping Table (TWGRCMAP)

This table stores the mapping between the channel SourceInfo, mode and database packages. This allows channels to retrieve data from Banner.

Mode is another concept in the Banner channel framework. The mode is passed to Banner to provide further information on what is to be retrieved. It is also used by the XML style sheet to transform the Banner information into the channel. The view can be changed by certain events while you remain in the same mode. For example, if you perform a search (one view), you will see the results (another view) while still in the same mode.

Note: This table is being delivered without any data. The script that will load the table is being delivered with Luminis Channels for Banner.

Column Name	Null?	Type	Column Comment
TWGRCMAP_NAME	Not null	VARCHAR2(50)	The data source name used when the channel was published.
TWGRCMAP_MODE	Not null	VARCHAR2(1)	This represents the different modes the channel will support, such as default or edit.
TWGRCMAP_VIEW	Not null	VARCHAR2(1)	This will show a different view from within the mode of the channel.
TWGRCMAP_PROCEDURE	Not null	VARCHAR2(255)	The name of the database package that will be called from the channel for the given mode and view. It will return XML.
TWGRCMAP_ACTIVITY_DATE	Not null	DATE	The date the record was created or last updated.
TWGRCMAP_USER_ID	Not null	VARCHAR2(30)	The database user who created or last updated the record.

Web Tailor Repeating Information Text Table (TWGRINFO)

This table contains user-defined text items that appear at various places in the Self-Service pages. You can provide instructions, background information, etc. with Info Text.

Column Name	Null?	Type	Column Comment
twgrinfo_name	Not null	VARCHAR2(120)	Name of associated page.
twgrinfo_label	Not null	VARCHAR2(20)	Short label, used to select which set of text items to print for a page. Default is DEFAULT.
twgrinfo_sequence	Not null	NUMBER	Sequence number for this text item.
twgrinfo_image		VARCHAR2(120)	Image associated with this text - overrides programmatic choice of image.
twgrinfo_text		VARCHAR2(2000)	Text to be displayed on Web page when this item is selected.
twgrinfo_comment		VARCHAR2(120)	Comment about this text item
twgrinfo_activity_date	Not null	DATE	Last date modified.
twgrinfo_source_ind		VARCHAR2(1)	Indicates if the row is Baseline (B) or Local (L). The default value is B.

Web Tailor Repeating Menu Item Table (TWGRMENU)

This table contains detail information about menu items. TWGBWMNU is its parent table.

Column Name	Null?	Type	Column Comment
twgrmenu_name	Not null	VARCHAR2(120)	Unique name of the menu or menu item. It is associated with one of the functional areas of the system.
twgrmenu_sequence	Not null	NUMBER	Sequence number for appearance on toolbar from top to bottom
twgrmenu_url_text		VARCHAR2(120)	The text string to be used as the hyperlink.
twgrmenu_url	Not null	VARCHAR2(120)	Text or expression that represents URL to link to when menu item is selected.
twgrmenu_url_image		VARCHAR2(120)	Name of image that can be printed with or in place of the text hyperlink, as found in TWGBIMAG.
twgrmenu_url_desc		VARCHAR2(2000)	Optional informational text that will be displayed below the link in the menu.
twgrmenu_image		VARCHAR2(120)	Name of image, as found in TWGBIMAG.
twgrmenu_enabled		VARCHAR2(1)	Menu item enabled switch.

Column Name	Null?	Type	Column Comment
twgrmenu_db_link_ind	Not null	VARCHAR2(1)	A value of Y indicates that this link is to a PL/SQL stored procedure; N indicates a static HTML link.
twgrmenu_submenu_ind	Not null	VARCHAR2(1)	Sub Menu item indicator. A value of Y indicates that it is a Sub menu; N indicates that it is not.
twgrmenu_target_frame		VARCHAR2(120)	Obsolete.
twgrmenu_status_text		VARCHAR2(120)	Text to display in the status bar when cursor is over this link.
twgrmenu_activity_date	Not null	DATE	Last date modified.
twgrmenu_source_ind		VARCHAR2(1)	Indicates if the row is Baseline (B) or Local (L). The default value is B.

Web Tailor Login Return Code Repeating Table (TWGRRETC)

This table contains the names of the packages associated with the login return locations specified in the TWGBRETC table.

Column Name	Null?	Type	Column Comment
twgrretc_return_code	Not null	VARCHAR2(30)	Login return code.
twgrretc_assoc_package		VARCHAR2(12)	PL/SQL Package name associated with this return code.
twgrretc_assoc_module		VARCHAR2(3)	Module associated with this return code.

Column Name	Null?	Type	Column Comment
twgrretc_activity_date	Not null	DATE	Date last modified.

Web Tailor User Roles Repeating Table (TWGRROLE)

This table defines roles that have been explicitly assigned to a user. Other roles are assigned dynamically when they logon based on a variety of different records throughout Banner.

Column Name	Null?	Type	Column Comment
twgrrole_pidm	Not null	NUMBER(8,0)	User PIDM.
twgrrole_role	Not null	VARCHAR2(30)	Role associated with this user (from Chevrolet).
twgrrole_activity_date	Not null	DATE	Date last modified.

Channel/Self-Service Preferences Table (TWGRUPRF)

This table stores institution-wide and user preferences for channels.

Note: This table was delivered without any data. The script to load the table was delivered with Luminis Channels for Banner.

Column Name	Null?	Type	Column Comment
TWGRUPRF_NAME	Not null	VARCHAR2(50)	The channel SourceInfo name.
TWGRUPRF_GROUP	Not null	VARCHAR2(30)	The group to which this preference belongs.

Column Name	Null?	Type	Column Comment
TWGRUPRF_KEY	Not null	VARCHAR2(30)	The key that will be used to identify the preference within the group.
TWGRUPRF_PREF_USER	Not null	VARCHAR2(30)	For delivered preferences, this is BASELINE . If a user customizes the preference, their ID will be stored here instead.
TWGRUPRF_SUB_ID	Not null	VARCHAR2(30)	Subscriber ID for the channel.
TWGRUPRF_DESC	Not null	VARCHAR2(255)	Description of the preference.
TWGRUPRF_VALUE	Not null	VARCHAR2(255)	Value for the preference. For BASELINE users this will be the default preference and for other users this will be a user-edited procedure.
TWGRUPRF_ACTIVITY_DATE	Not null	DATE	The date the record was created or last updated.
TWGRUPRF_USER_ID	Not null	VARCHAR2(30)	Banner ID of the person who created or last updated the record.
TWGRUPRF_EDIT_IND	Not null	VARCHAR2(1)	Indicates if the preference can be edited by the end users. Valid values are: Y - the preference can be edited by end users N - the preference cannot be edited; it will not be displayed to users in the channel.

Column Name	Null?	Type	Column Comment
TWGRUPRF_DISPLAY_TYPE		VARCHAR2(10)	HTML control used to display this preference, e.g., check box, option button, etc.
TWGRUPRF_SORT_ORDER		NUMBER	Order in which the preference should be displayed to the end users.

Web Tailor Menu Roles Table (TWGRWMRL)

This table contains records that define the roles a user must have to access a particular menu or page.

Column Name	Null?	Type	Column Comment
twgrwmrl_name	Not null	VARCHAR2(120)	Name of procedure or menu from TWGBWMNU.
twgrwmrl_role	Not null	VARCHAR2(30)	User role associated with this menu or procedure.
twgrwmrl_activity_date	Not null	DATE	Last date modified.
twgrwmrl_source_ind		VARCHAR2(1)	Indicates if the row is Baseline (B) or Local (L). The default value is B.

Web Tailor User Parameters Table (TWGRWPRM)

This table contains temporary attribute/value combinations for a specific user. These combinations provide session state context between database procedure calls.

Column Name	Null?	Type	Column Comment
twgrwprm_pidm	Not null	NUMBER(8,0)	User PIDM.
twgrwprm_param_name	Not null	VARCHAR2(10)	Parameter name.
twgrwprm_param_value		VARCHAR2(30)	Parameter value.
twgrwprm_activity_date	Not null	DATE	Last date modified.

Web Tailor Module Validation Table (TWTVMODU)

This table contains definitions of all the Self-Service modules. You should only update the information in the table, not insert records into it or delete records from it.

Column Name	Null?	Type	Column Comment
twtvmodu_code	Not null	VARCHAR2(3)	Module code - one exists for each Web product.
twtvmodu_desc	Not null	VARCHAR2(120)	Description of the module.
twtvmodu_global_menu		VARCHAR2(120)	Menu (twgbwmnu) to be displayed at the bottom of every page associated with this module.

Column Name	Null?	Type	Column Comment
twtvmodu_header_caps_on		VARCHAR2(1)	A value of Y indicates that the Header Text in this module is printed with <code>twbkfmt.P_HeaderCaps</code> . In Release 5.2, this was replaced by values on the page-level CSS.
twtvmodu_display_exit_ind	Not null	VARCHAR2(1)	A value of Y indicates that the Exit/Logout button is to be printed on pages within this module.
twtvmodu_curr_release_no		VARCHAR2(30)	Current release number for this module. Will override setting in <code>twbkwbis</code> , but not value passed to <code>P_CloseDoc</code> .
twtvmodu_activity_date	Not null	DATE	Date last modified.
twtvmodu_css_url		VARCHAR2(255)	URL where the module-level cascading style sheet is stored.
twtvmodu_help_image		VARCHAR2(120)	Obsolete.
twtvmodu_exit_image		VARCHAR2(120)	Obsolete.
twtvmodu_menu_image		VARCHAR2(120)	Obsolete.
twtvmodu_back_image		VARCHAR2(120)	Obsolete.
twtvmodu_help_css		VARCHAR2(255)	Indicates where the module-level Help CSS is stored.
twtvmodu_help_url		VARCHAR2(255)	Holds the URL of the module-level static HTML page.

Web Tailor User Role Validation Table (TWTVROLE)

This validation table lists the valid roles that can be granted to a user or assigned to a menu item.

Column Name	Null?	Type	Column Comment
twtvrole_code	Not null	VARCHAR2 (30)	Role code.
twtvrole_desc	Not null	VARCHAR2 (60)	Description of the user role.
twtvrole_activity_date	Not null	DATE	Date last modified.
twtvrole_user_defined_ind	Not null	VARCHAR2(1)	A value of Y indicates that the role is defined on twgrrrole; N indicates that it is data-driven from GOVROLE.
twtvrole_time_out	Not null	NUMBER	TIME OUT: Period of inactivity on the web which will cause a timeout (in minutes). This value overrides TWGBWRUL_TIME_OUT if it is a longer period of time.

Web Tailor Packages

This chapter provides reference information on the packages that support Web Tailor processing.

Overview

This section provides a technical summary of the Web Tailor packages that are used in Web processing. Where appropriate, cursors, variables, constants, functions and procedures are documented, and a brief description is supplied.

This information is provided to assist technical staff in understanding the technical processing in Web Tailor. In addition, any site-specific Web development may be able to use delivered Banner™ Self-Service packages, eliminating the need to write new code in some cases.

TWBKAUSR

This package, designed to handle alternate user IDs, maps the user ID on the SPRIDEN table to an external user.

Procedures

`p_main_wrapper` - Contains the URL as `<servername><adminDAD>\twbklogn.P_Login`
`p_web_login` - Contains `SSO_USER`, `SSO_PSWD`, `SSO_LUM_USER` and the URL as `<servername><adminDAD>\twbklogn.P_Login`

TWBKTABS

This package contains definitions of table types to be used when processing Web Tailor data.

TWBKSELS

This package contains cursors, simple procedures, and functions that retrieve data contained in Web Tailor tables.

Note: In most cases Web Tailor calls are made to a given procedure, for example, `p_printanchor`. Many of those procedures call a corresponding function which actually does the work, for example, `f_printanchor`. Procedures begin with `p_` and functions begin with `f_`.

These procedure/function pairs were created so that applications that require the HTML to be constructed in a buffer and printed later can call the function directly, while applications that need the output to be generated immediately can call the procedure.

Cursors

`twgbwses_row_by_pidm_c` - Retrieves rows from TWGRWPRM for a specified TWBRWPRM_PIDM.

`twgrwprm_row_by_pidm_name_c` - Retrieves rows from TWGRWPRM for a specified TWBRWPRM_PIDM and TWBRWPRM_PARAM_NAME.

`twgrwprm_row_pidm_like_name_c` - Retrieves a row from table TWGRWPRM based on PIDM and value.

`twbrwprm_row_by_name_value_c` - Retrieves rows from TWGRWPRM for a specified TWBRWPRM_PARAM_NAME and TWBRWPRM_PARAM_VALUE.

Functions

`f_generate_random_id` - Generates a six-character random ID from the value submitted.

`f_get_param_value` - Retrieves the TWBRWPRM_PARAM_VALUE for a specified TWBRWPRM_PARAM_NAME and TWBRWPRM_PIDM.

`f_get_pidm_from_twbrwprm` - Retrieves the TWBRWPRM_PIDM from TWGRWPRM for a specified TWBRWPRM_PARAM_VALUE.

TWBKMODS

This package contains procedures and functions that insert new records into Web Tailor tables, update existing records in Web Tailor tables, and delete existing records from Web Tailor tables.

`p_del_twgrwprm_pidm_like_name` - Deletes existing records from TWGRWPRM, where TWGRWPRM_PIDM equals the specified pidm and TWGRWPRM_PARAM_NAME is the specified parameter name.

`p_delete_twgrwprm` - Deletes existing records from TWGRWPRM for a specified TWGRWPRM_PIDM and, optionally, up to two (2) TWGRWPRM_PARAM_NAME values.

`p_delete_twgrwprm_pidm_name` - Deletes existing records from TWGRWPRM for a specified TWGRWPRM_PIDM and TWGRWPRM_PARAM_NAME.

`p_insert_twgbwses` - Inserts new records into TWGBWSES for a specified TWGBWSES_PIDM with the values provided.

`p_insert_twgrwprm` - Inserts new records into TWGRWPRM for a specified TWGRWPRM_PIDM with the values provided.

`p_update_twgbwses` - Updates existing records in TWGBWSES for a specified TWGBWSES_PIDM with the values provided.

`p_update_twgrwprm` - Updates existing records in TWGRWPRM for a specified TWGRWPRM_PIDM with the values provided.

TWBKFUNC

This package contains procedures and functions that process data in Web Tailor tables.

Procedures

`p_del_twgrwprm_by_pidm_name` - Deletes TWGRWPRM rows for a specified TWGRWPRM_PIDM, where at least part of TWGRWPRM_PARAM_NAME matches the specified parameter name. Uses the Oracle LIKE operator and appends the wildcard matching character (%) to the end of the string to match.

`p_del_twgrwprm_by_value` - Deletes TWGRWPRM rows for a specified TWGRWPRM_PARAM_VALUE.

`p_get_twgrwprm_rows` - Retrieves TWGRWPRM rows for a specified TWGRWPRM_PIDM and TWGRWPRM_PARAM_NAME.

`p_store_twgrwprm_row` - Builds a parameter name from values submitted and updates an existing row in TWGRWPRM, or inserts a new row if one does not already exist.

`p_validate_session_id` - Validates the current webid (`value_in`) against the previously generated webid for the specified PIDM (`pidm_in`), and generates a new webid (`temp_id_in_out`) from the value submitted.

Functions

`f_build_param_name` - Builds a parameter name from values submitted to be used when retrieving, updating, inserting, or deleting TWGRWPRM rows.

`f_load_and_store_param_value` - Builds a parameter name, deletes existing rows with that name, builds a new parameter name, and updates an existing TWGRWPRM row or inserts a new row if one does not exist for the new parameter name.

TWBKBSF

This package contains the encryption procedures used in the Luminis Platform security routines.

`f_binary_to_integer`

```
f_decode_base64 (base64str IN VARCHAR2, pad IN CHAR DEFAULT '=') RETURN  
VARCHAR2;
```

Encodes the str value using the base64 encoding method as described above.

`f_decode_bits` - Decodes the value passed to the `base64str` parameter using the Base64 method described above.

`f_encode` - Encodes the following special characters so they can be passed through the Web server without being misinterpreted:

Value	Encoded Value
%	%25
+	%2B
sp	+
/	%2F
:	%3A
;	%3B
@	%40
&	%26
=	%3D
?	%3F
'	%27

```
f_encode_bits (binval IN VARCHAR2) RETURN VARCHAR2;
```

```
f_hex_to_integer
```

```
f_integer_to_binary
```

```
f_integer_to_hex
```

The following procedures are used within the encoding and decoding process to process sections of the source string.

```
p_encode_pos1
```

```
p_encode_pos2
```

The following two procedures use calls to the `UTL_RAW` package to emulate binary shifting of the string passed into the `rawin` parameter.

```
p_rightshift
```

```
p_leftshift
```

TWBKAUTH

This package contains the procedures which pass secured information in the Luminis Platform application.

Global variables

Global variables used to parse the security info in the CP global cookie.

```

type longvarchar2_tabtype is table of VARCHAR2(32767)
  index by binary_integer;
cp_endpos_tab          twbklibs.VARCHAR2_tabtype;
cp_startpos_tab       twbklibs.VARCHAR2_tabtype;
cp_value_tab          longVARCHAR2_tabtype;
cp_ent_value_tab      longVARCHAR2_tabtype;
cp_type_tab           twbklibs.VARCHAR2_tabtype;
cp_num_cookie_vals    number := 0;
cp_num_cookie_vals    number := 0;

```

Functions

F_CPAuthenticate - Takes in the user's ID and PIN, and returns a combination of the Web session id and pidm if the user is valid. If the user is not valid, the function will return a NULL.

F_FetchCPCookieval

F_FetchCPLastAccess

F_GetLastActivityTime - Takes the user's ID and returns the last activity time found in the Web session table (TWGBWSES). The format of the return string is as specified in the CP cookie-encoding spec. Returns NULL if the user is not found, or the user does not have a last activity time stored in the Web session table.

F_ReconstructCPCookie - Puts the CP cookie back together, using the PL/SQL tables above with the updated Web ID and user PIDM. This returns the plain text version of the cookie. It will be encoded later.

F_RetrieveRoles - Takes the user's ID, and passes back a comma-delimited string of CP-recognized role names that are valid for the specified user. If no roles are found, the function returns a NULL.

Procedure

P_ParseCPCookie - Takes the CP cookie value (after it has been decoded), splits it into sections based on the CP cookie specifications. The pieces are loaded into the PL/SQL tables (defined above) for later use.

Functions

`F_CPAuthenticate` - Takes in the user's ID and PIN, and returns a combination of the Web session id and pidm if the user is valid. If the user is not valid, the function will return a `NULL`.

`F_FetchCPCookieval`

`F_FetchCPLastAccess`

`F_GetLastActivityTime` - Takes the user's ID and returns the last activity time found in the Web session table (TWGBWSES). The format of the return string is as specified in the CP cookie-encoding spec. Returns `NULL` if the user is not found, or the user does not have a last activity time stored in the Web session table.

`F_ReconstructCPCookie` - Puts the CP cookie back together, using the PL/SQL tables above with the updated Web ID and user PIDM. This returns the plain text version of the cookie. It will be encoded later.

`F_RetrieveRoles` - Takes the user's ID, and passes back a comma-delimited string of CP-recognized role names that are valid for the specified user. If no roles are found, the function returns a `NULL`.

Procedure

`P_ParseCPCookie` - Takes the CP cookie value (after it has been decoded), splits it into sections based on the CP cookie specifications. The pieces are loaded into the PL/SQL tables (defined above) for later use.

TWBKLDAP

This package creates the LDAP Administration page. You will use this page to maintain the LDAP Settings Table (TWGBLDAP) because some of the columns in TWGBLDAP are encrypted so the information they contain cannot be viewed in SQL*Plus.

Procedures

`P_ModifyPgLDAP` - Displays a page showing information from the TWGBLDAP table.

<code>return_code</code>	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
<code>return_msg</code>	A message to be displayed on screen
<code>trans_type</code>	Transaction type indicator; U - update, C - create

*_in	Input parameters - will be null if successful. If not null, the update or insert failed, and the user must resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed.
P_SaveLDAP - Allows you to save changed information to the TWBKLDAP table.	
Warning! The values of the passwords will be encrypted when you save.	
trans_type	Transaction type indicator. Valid values: U - update, C - create
*_in	Fields from the data entry form that correspond to the database tables.

TWBKLOGN

This package contains the procedures and functions that authenticate user credentials.

Note: If you want to create a custom function, it is recommended that you copy one of the existing functions, modify it, and change the Web Tailor parameter LDAPFUNCTION to point to it.

This package calls the GB_THIRD_PARTY_ACCESS API for all inserts and updates to the GOBTPAC table.

Functions

F_ENC_DATA - Performs encryption using the Oracle DBMS_OBFUSCATION package, and returns the encrypted string.

F_LDAP_BANNERSEARCH - Returns the mapping from the GOBTPAC table. Use this function if the LDAP user is mapped to Banner Self-Service by storing the LDAP user ID in the GOBTPAC_LDAP_USER column in the GOBTPAC table.

Warning! When mapping an LDAP user ID on the GOATPAD form, be sure to assign a different LDAP ID for each Banner ID. They must be unique.

F_LDAP_CPSEARCH - This function is used if your institution stores user credentials on an LDAP server instead of maintaining them on the GOBTPAC table. It performs an LDAP search to retrieve the Banner ID. In addition to mapping LDAP to Banner Self-Service, it also manipulates the returning string to remove extraneous text from the end of it.

F_LDAP_CUSTOMSEARCH - Performs an LDAP search to retrieve the Banner ID. It returns a string exactly as it is. Use this function if the LDAP user is mapped to Banner Self-Service by storing Self-Service IDs as an attribute in LDAP.

If a GOBTAC record is not found (and LDAP is being used as the authentication source), one will be inserted automatically for the first login. The Terms of Usage indicator information (selected or cleared) will be included in this record. However, this will not store PIN information.

Procedures

`P_ValLogin` - Procedure to validate the login.

<code>sid</code>	Student ID
<code>pin</code>	Personal Identification string
<code>newpin</code>	New PIN number
<code>verpin</code>	Verification of new PIN number

TWBKINTE

This package contain the procedures which perform the necessary steps to allow end users of the Banner Web systems to jump to another Web Tailor-based system without having to authenticate themselves again, and back again when backing out.

Procedures (used on remote side)

`P_ForwardFromBanner` - Used to check `jumpid` to make sure is valid, update `sessid` and re-route user to the correct initial screen.

`P_ReturnToBanner` - Used to send the end user back to Banner WT system.

Procedures (used on Banner side)

`P_ReturnFromOther` - Used to re-establish security on the Banner system when user returns back to Banner from other system from executing `P_ReturnToBanner`

`P_ForwardToOther` (`systemname` VARCHAR2) - Used to create a `jumpid` cookie and redirect to another Web Tailor-based system and execute `P_ForwardFromBanner` on that system

TWBKLIST

The TWBKLIST package was introduced in Release 7.1 to close a potential security hole.

Some Self-Service procedures exist in executable packages that a user previously could call directly in a URL without being logged in. This caused a security vulnerability through which users could potentially access/update data that they would normally not be able to access.

This package solves the security issue by preventing users from calling a procedure directly without being logged in.

Procedures

`P_main` - Throws an exception if a procedure is executed directly and is not defined in the `TWGBWMNU` table. The process looks at an environment variable to determine which procedure is being called. This process will need to be called from `Before` procedure in the DAD configuration.

TWBKFRMT

This package provides general HTML formatting functions, which are used by other Web Tailor packages.

Functions

`F_ConvertTextareaTags` - Converts any greater than or less than signs (`<` `>`) in the specified string to `<` and `>` respectively. This is used in textarea value strings, to ensure that the HTML tags stored in a database field are displayed on the page.

`F_ConvertURLSpaces` - Converts any spaces in the specified string to `%20`, which is interpreted by a Web server as a space. Used to make sure that URLs with query strings do not contain any spaces.

`f_display_ssb_field` - Conditionally displays fields based on GORDMSK rules established for the `**SSBMASKING` object and the specific package and item. For instructions on using this masking feature, see "Masking International Person Information Fields" in Chapter 4 of the *Banner General Data Security Handbook*.

`F_EncodeURL` - Encodes the following characters in the input string: `<space>` `?` `/` `;` `:` `@` `=` `&`

`f_err_msg_remove_delim_tbl` - Formats and displays any exceptions from a procedure's API call.

`f_Escape` - converts suspicious strings to ordinary text before passing them to the browser to be displayed

`F_FormDate_txt`

`F_FormDateText` - Prints a form text field with the length set to the current length of the date input format. Prints the date input format to the right of the field to instruct users.

`F_FormHidden` - replacement for `htf.formhidden`

`F_FormText` - Generates the `<INPUT>` tag by calling the Oracle Web Toolkit function `htp.FormText`. It may also make some adjustments to the input field width.

`F_FormLabel` - Prints `cvalue` as a label, applying `idname` as the ID attribute.

- **Visible** takes two values, `Visible` and `Invisible`, to allow a label to be output but not displayed by the browser. This is occasionally required when a label is necessary for screen readers but not for display. The invisible attribute is controlled by the style sheet classes `fieldlabeltextinvisible`. Visible text is controlled by the style sheet class `fieldlabeltext`.
- `Printrequired` optionally outputs `required` image as defined on the Web Tailor Customize Global User Interface Options page.

`F_FormText` - Produces the `<INPUT>` tag to provide text fields for data entry.

`F_HeaderCaps` - Prints a large bold heading, where `string_in` is converted to all capital letters, with initial capital letters two sizes larger than the rest.

`F_ImageLink` - Returns the HTML text that displays the link specified, using the image as the hotspot for the link.

<code>url</code>	The URL of the new hyperlink
<code>imagename</code>	The high-level image name, from <code>TWBKIMAG_NAME</code>
<code>anchorname</code>	Placeholder name to be associated with the anchor tag (prints <code><A NAME="anchorname"...</code>)
<code>target</code>	Frame that the link should open in (e.g., <code>_top</code> or <code>_self</code>)
<code>cattributes</code>	Any additional components to be included in the anchor tag

`F_PrintAnchor` - Produces `<A>` `` tags to create a hyperlink.

`F_PrintBold` - Prints text with the bold attribute. This is a deprecated function and should not be used for generating accessible HTML.

`F_PrintBullet` - Returns the text of the HTML `IMG` tag for printing the bullet for the specified page. Checks the `TWBKWMNU` record to find the specified bullet name.

`F_PrintDivider` - Returns the text of the HTML `IMG` tag for printing the divider for the specified page. Checks the `TWBKWMNU` record to find the specified divider name.

`F_PrintHeader` - Produces the `<H(csize)>` tag. Parameters are the same as the Oracle Web toolkit `HTF.header`, but if `cheader`, the header text, is null, it returns `NULL` and no `<H>` tag is produced at all.

`F_PrintImage` - Returns the text of the HTML `IMG` tag for printing the specified image.

<code>imagename</code>	The high-level image name, from <code>TWBKIMAG_NAME</code>
<code>cismap</code>	<code>ISMAP</code> if this image is a server-side imagemap

cusemap	Specifies the client-side imagemap to be used (make sure that the name of the map is prefaced by a hash mark '#')
---------	---

`F_PrintText` - Prints text inside `` `` tags using the CSS class defined by `class_in`.

`F_Object` - Formats an object tag

`F_ObjectOpen` - Formats an object open tag.

`F_ObjectClose` - Formats an object close tag.

`F_PrintRequired` - Prints a marker (the image in TWBKIMAG called `web_required`) that specifies a required field on a form using CSS class `fieldrequiredtext`. If `web_required` is not defined, it defaults to `*`.

`F_PrintSpacer` - Returns the text of an HTML `IMG` tag that uses a single-pixel transparent gif file (`TWBKIMAG_NAME='GeneralSpacer'`) to create a spacer.

`F_TableClose` - Produces the `</TABLE>` tag and restores previous table type if defined.

`F_TableData` - Forms the basis for many other table data functions. It generates the `<TD>` or `<TH>` tags with the appropriate style sheet classes based on the input parameters.

It is sensitive to the type of table, implemented in the package global variable `current_table_type`. `current_table_type` is initialized when `twbkfrmt.f_tableopen` is called, and reset when `twbkfrmt.f_tableclose` is called.

The style sheet class is defined by the type of table and the datatype parameter, as follows:

Current Table Type	Class Attribute
DATAENTRY	<code>class='de<datatype>'</code>
NONTABULAR	<code>class='nt<datatype>'</code>
DATADISPLAY	<code>class='dd<datatype>'</code>
PLAIN	<code>class='pl<datatype>'</code>
MENUPLAIN	<code>class='mp<datatype>'</code>
INFO	<code>class='in<datatype>'</code>
DATAWITHBORDER	<code>class='db<datatype>'</code>

If the current table type is `DATAENTRY` and the datatype is `HEADER`, the `TH` tag is returned with the class defined as `CLASS='deheader'`.

In addition, a `SCOPE=` attribute is defined when `current_table_type` is `DATAENTRY`, `DATADISPLAY`, `NONTABULAR`, or `DATAWITHBORDER`:

- If datatype is `TITLE` then scope is set to `colgroup`.
- If datatype is `HEADER` then scope is set to `col`.
- If datatype is `LABEL` then scope is set to `row`.

cvalue	The value for the VALUE attribute
cdp	The value for the DP (decimal point) attribute. Passed as is.
crowspan	Value of the ROWSPAN attribute
ccolspan	Value of the COLSPAN attribute
cattributes	Other attributes, passed as is.
datatype	TITLE, HEADER, LABEL generate TH tag if current_table_type is DATADISPLAY. Otherwise, a TD tag is generated

`F_tableDataOpen` - Generates the `<TD>` tag, with class and alignment controlled by style sheet classes in the same way as `f_tableopen`. It keeps track of table nesting levels.

`F_TableDataClose` - Generates the `</TD>` tag and keeps track of table nesting levels.

`F_TableDataDead`

`F_TableDataDeadClose`

`F_TableDataHeader`

`F_TableDataHeaderOpen`

`F_TableDataHeaderClose`

`F_TableDataHighlight`

`F_TableDataHighlightOpen`

`F_TableDataHighlightClose`

`F_TableDataLabel`

`F_TableDataLabelOpen`

`F_TableDataLabelClose`

`F_TableDataSeparator`

`F_TableDataSeparatorOpen`

`F_TableDataSeparatorClose`

`F_TableDataWhite`

`F_TableDataWhiteOpen`

`F_TableDataWhiteClose`

`F_TableHeader`

`F_TableOpen`

`F_TableRowClose`

`F_TableRowOpen`

Procedures

P_FormDateText

P_FormHidden - replacement for `http.formhidden`.

P_FormLabel - Uses `F_formlabel` to print an item inside a `<LABEL>` tag, optionally using a CSS to control visibility, and if `printrequired = Y`, then prints the system-level required-field indicator icon and text.

P_FormRadioRow - Place a Labeled Radio Button Field (Yes/No buttons) on a page, check for missing values if a required field. If it is not a required field, and a string is passed via the `no_resp_flag` parm, the No Response button is also displayed.

P_FormSelectDay - Place a pull-down list containing the days of a month on a page.

P_FormSelectMonth - Place a pull-down list containing the months of the year on a page.

P_FormText

P_FormTextRow - Place a Labeled Text Field on a page, check for missing values if a required field.

P_HeaderCaps - Prints a large bold heading, where `string_in` is converted to all capital letters, with initial capitals two sizes larger than the rest.

P_ImageLink - Prints the HTML that displays the link specified, using the image as the hotspot for the link.

<code>url:</code>	The URL of the new hyperlink.
<code>imagename</code>	The high-level image name, from <code>TWBKIMAG_NAME</code> .
<code>anchorname</code>	Placeholder name to be associated with the anchor tag (prints <code></code>).
<code>target:</code>	Frame that the link should open in (e.g., <code>_top</code> or <code>_self</code>).
<code>cattributes:</code>	Any additional components to be included in the anchor tag.

P_PrintAnchor

P_PrintApiMessages - Displays API-generated error messages.

P_PrintBold

P_PrintBullet - Prints the HTML IMG tag for the bullet on the specified page. Checks the `TWBKWMNU` record to find the specified bullet name.

P_PrintDivider - Prints the HTML IMG tag for the divider on the specified page. Checks the `TWBKWMNU` record to find the specified divider name.

P_PrintHeader

`P_PrintImage` - Prints the HTML that displays the image specified.

<code>imagename</code>	The high-level image name, from <code>TWBKIMAG_NAME</code>
<code>cismap</code>	<code>ISMAP</code> if this image is a server-side image map.
<code>cusemap</code>	Specifies the client-side imagemap to be used (make sure that the name of the map is prefaced by a hash mark.#)

`P_PrintMenu` - Prints a menu with items from `TWGRMENU` (similar to the `GORWMIT` table -- this procedure prints the links by themselves, and does not include an open and close document portion. This way, one procedure can print both bottom links and full-page menus). This procedure displays header links as tabs instead of hyperlinks.

<code>name</code>	Name of the menu, from <code>TWBRMENU_NAME</code>
<code>display_type</code>	Current valid values are: <code>F</code> - Footer, or bottom, links. This prints the links in rows, with a certain number of links on each row. They are printed in a table with centered text alignment. <code>B</code> - Body, or full-page, menu. The links are printed in a bulleted format
<code>font_size</code>	If the size of the links should be different from the default, send the new size here (e.g., <code>font_size='-2'</code> to have the text display two sizes smaller than the surrounding text)
<code>num_in_row</code>	By default, for a footer menu, there are 3 links on each row. This is the override for that number.
<code>validate_links</code>	By default, the <code>twbkwbis.f_validlink</code> call is made before printing a menu item. If the menu is generated within the Student Admissions module, the user will not be authenticated with normal Web Tailor security. When this flag is <code>FALSE</code> , links are printed as is.
<code>map_title</code>	Header and footer link menus generated by <code>p_printmenu</code> will be enveloped in a <code>MAP</code> tag. The <code>map_title</code> parameter will accept a value that will be assigned to the title attribute on the <code>MAP</code> tag.
<code>ccaption</code>	Full page menus generated by <code>p_printmenu</code> will be enveloped in an HTML table. The <code>ccaption</code> parameter will accept a value that

will be assigned to the caption attribute on the TITLE tag.

`P_PrintMessage` - Prints a message with either the error image, warning image or no image. It automatically displays API-generated exceptions by retrieving the API error message string and displaying it at the top of the Web page (instead of propagating an unhandled exception to the page).

<code>message</code>	The message to be printed.
<code>message_type</code>	Determines how <code>P_printmessage</code> outputs the message: 1 - Error 2 - Warning 3 - Note The default value is null.

`P_PrintMessages` - Removes delimiters from API error messages before they are displayed.

`P_PrintRequiredMsg` - Prints a message explaining that required fields are marked with a check.

`P_PrintStrong`

`P_PrintText` - Prints HTML text with global font face settings. Default size is the browser DEFAULT (3).

`P_Prn` - replacement for `htf.prn`

`P_StoreApiMessages` - Stores error message strings in `api_error_txt`.

`P_TableClose`

`P_TableData`

`P_TableDataOpen`

`P_TableDataClose`

`P_TableDataDead`

`P_TableDataDeadOpen`

`P_TableDataDeadClose;`

`P_TableDataHeader`

`P_TableDataHeaderOpen`

`P_TableDataLabelClose;`

`P_TableDataSeparator`

`P_TableDataSeparatorOpen`

`P_TableDataSeparatorClose`

`P_TableDataWhite`

P_TableDataWhiteOpen
 P_TableDataWhiteClose
 P_TableHeader
 P_TableRowClose
 P_TableRowOpen
 P_TableDataHeaderClose
 P_TableDataHighlight
 P_TableDataHighlightOpen
 P_TableDataHighlightClose
 P_TableDataLabel
 P_TableDataLabelOpen
 P_TopLeftBorder - Prints top and left borders for validation form option pages.

border_type	<p>There are two types of boxes used in Web Tailor: C - Create and U - Update</p> <p>These boxes are outlines, created with tables, and are different colors depending on which type of box it is. The default colors can be changed in the global variables declared in the TWBKWBIS package.</p>
label	Optional text to be printed in the upper border of the box.

Tab-style navigation

Three procedures in TWBKFRMT support tab-style navigation at the page level. This capability was added with Web Tailor Release 7.2.

p_init_tab - Initializes a web page that uses tab navigation.

p_set_tab - Creates each tab.

p_close_tab - Closes a tab.

TWBKGLIB

The procedures in this package are borrowed from GOKLIBS. Because Web Tailor does not require a Banner database, the GOKLIBS package will not always be present, and cannot be referenced by other Web Tailor procedures and functions.

Cursors

GobtpacC - Selects a row from the GOBTPAC third-party access table.

Functions

F_IsDate - Determines if input is a valid date.

F_IsNumber - Determine if input is a valid number.

F_PinLength - Returns the Web Tailor-defined PIN length.

F_PinNextExpDate - Returns today + Web Tailor-defined PIN expiration number of days.

F_PinReused - Determines if the input PIN has already been used.

F_StrAllDigits - Determines if a string contains all numeric digits.

TWBKIMAG

The procedures in this package print the Graphic Elements pages on the Web, so end users can update the data in the TWGBIMAG table.

Procedures

P_ModifyPgImage - Page that allows the user to enter and change information in the TWGBIMAG table. Uses a call to a javascript procedure to do data validation before the form is submitted.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
return_msg	A message to be displayed on screen.
trans_type	Transaction type indicator. U for update, C for create
*_in	Input parameters - will only be not null if the update or insert failed and the user is asked to

resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed.

P_OptionPgImage - Page that gives users the option of creating or updating an existing image.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
-------------	---

P_SaveImage - Updates or inserts a row into the TWGBIMAG table, depending on user inputs and the transaction type.

trans_type	Transaction type indicator. 'U' for update, 'C' for create.
------------	---

*_in	Fields from the data entry form that correspond to the database tables.
------	---

TWBKJSCR

This package contains general PL/SQL functions and procedures which generate javascript code.

Functions

F_JScriptLinkPreviewColor - Function which prints the HTML anchor tag that links to the Preview Color window.

field_name	Name of HTML form field that contains hexadecimal RGB color code to be previewed in Preview Color window.
------------	---

F_JScriptLinkPreviewImage - Prints the HTML anchor tag that links to the Preview Image window.

field_name	Name of HTML form field that contains image name to be previewed in Preview Image window.
------------	---

Procedures

P_JScriptClose - Prints a beginning HTML SCRIPT tag, specifying the start of javascript code.

P_JScriptLinkPreviewColor - Prints the HTML anchor tag that links to the Preview Color window.

field_name	Name of HTML form field that contains hexadecimal RGB color code to be previewed in Preview Color window.
------------	---

P_JScriptLinkPreviewImage - Procedure which prints the HTML anchor tag that links to the Preview Image window.

field_name	Name of HTML form field that contains image name to be previewed in Preview Image window.
------------	---

P_JScriptOnSubmit - Creates a function that shows a message on the screen if the user selects the **Submit** button twice.

P_JScriptOpen - Prints an ending HTML SCRIPT tag, specifying the end of javascript code.

P_JScriptPreviewColor - Prints a javascript function that opens the Preview Color window. This procedure opens the file called /wtlhelp/twbhjclr.htm, under the Web server document root.

P_JScriptPreviewImageClose - Releases the lock on the TWBKIMAG which is created by the P_JScriptPreviewImageOpen procedure.

P_JScriptPreviewImageOpen - Prints a javascript function that opens the selected image in a Preview Image window. Creates an array which contains the high-level image names (from TWBKIMAG) and their associated URLs, so that the correct image path is previewed when the user selects the Preview Image link. Places a lock on the TWGBIMAG table, so that the image array matches the order and contents of the image pull-down lists on the Web page.

P_JScriptWindowOpen - Prints javascript function which opens a smaller browser window, for displaying help pages.

TWBKLIBS

This package, the Web Tailor Library package, declares the WWW Banner General cursors and exceptions.

Records

twgbwses_rec	twgbwses%rowtype;	Web sessions
--------------	-------------------	--------------

twgbwrul_rec	twgbwrul%rowtype;	Web rules configuration row
twgbwmnu_rec	twgbwmnu%rowtype;	Web Menus row
twgrinfo_rec	twgrinfo%rowtype;	Web Info row
twgrwmrl_rec	twgrwmrl%rowtype;	Web menu roles row
twtvmodu_rec	twtvmodu%rowtype;	Web modules row
twgbglui_rec	twgbglui%rowtype;	Global user interface parameters

Types

```

TYPE VARCHAR2_tabtype is table of VARCHAR2(1000)
  index by binary_integer;
Generic VARCHAR2 table type.
TYPE error_msg_tabtype is table of VARCHAR2(200)
  index by binary_integer;
Error message table type;

```

Variables

error_msg_table error_msg_tabtype;	Error message table
date_input_fmt VARCHAR2(8);	Format for date input on the Web
display_desc_names boolean;	Gets updated from twbkwrul_disp_admin_disc_ind upon package initialization.

Cursors

GetSessionC - Fetches Web session row.

GetMenuC - Fetches Web menu or procedure row.

GetMItemC - Fetches menu item row.

GetInfoC - Fetches information text row.

GetModuleC - Fetches Web module row.

GetWmrlC - Fetches Web menu roles row.

Exceptions

```
general_error_num      constant number := -20000;
  general_error        exception;
GetMenuError_num      constant number := -20100;
  GetMenuError         exception;
GetUserError_num     constant number := -20101;
  GetUserError        exception;
GetSessionError_num  constant number := -20102;
  GetSessionError     exception;
NewPinMismatch_num   constant number := -20103;
  NewPinMismatch      exception;
OldPinMismatch_num   constant number := -20104;
  OldPinMismatch      exception;
UserDisabled_num     constant number := -20105;
  UserDisabled        exception;
UserProhibited_num   constant number := -20106;
  UserProhibited      exception;
PinNotAllDigits_num  constant number := -20107;
  PinNotAllDigits     exception;
PinNotFullLength_num constant number := -20108;
  PinNotFullLength    exception;
PinNotNew_num        constant number := -20109;
  PinNotNew           exception;
PageEnabledError_num constant number := -20110;
  PageEnabledError    exception;
PageRoleError_num    constant number := -20111;
  PageRoleError       exception;
UserDead_num         constant number := -20112;
  UserDead            exception;
MissingGubinst_num   constant number := -20113;
  MissingGubinst      exception;
MissingTwbkwrul_num  constant number := -20114;
  MissingTwbkwrul     exception;
MissingGobtpac_num   constant number := -20115;
  MissingGobtpac      exception;
PageNotFoundError_num constant number := -20116;
  PageNotFoundError   exception;
ldapattribmaperror_num constant number := -20117;
  ldapattribmaperror  exception;
ldapuniqueerror_num  constant number := -20118;
  ldapuniqueerror     exception;
ldapusernotfound_num constant number := -20119;
  ldapusernotfound    exception;
```

TWBKMENU

This package allows users to add, delete, update, or reorder menu items.

P_ModifyPgMenuItem - Page that allows the user to enter/change information in the TWGRMENU table. Uses a call to a javascript procedure to do data validation before the form is submitted.

<code>return_code</code>	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
<code>return_msg</code>	A message to be displayed on screen.
<code>trans_type</code>	Transaction type indicator; U - update, C - create.
<code>*_in</code>	Input parameters - will be null if successful. If not null, the update or insert failed, and the user must resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed.

P_OptionPgMenuItem - Page that gives user option of creating or updating an existing set of menu entries (repeating records).

P_ReorderMenuItem - Back-end process which reorders the menu items for the specified menu, then returns to the Reorder Menu Items page.

<code>name_in</code>	Name of the menu, from TWBRMENU_NAME
<code>orig_seq_tab</code>	Table of original sequence numbers as printed on the Reorder Menu Items page.
<code>new_seq_tab</code>	Table containing the values of menu item sequence number pull-down lists on the page.

P_ReorderPgMenuItem - Page that displays all current menu items, gives links to update each item, and an interface for reordering elements.

<code>name_in</code>	Menu name from TWBRMENU_NAME
<code>return_msg</code>	Return message to be printed.
<code>return_code</code>	Error return code.

P_SaveMenuItem - Updates or inserts a row into the TWGRMENU table, depending on user inputs and the transaction type.

trans_type	Transaction type indicator. Valid values are: U - update, C - create.
*_in	Fields from the data entry form that correspond to the database tables.

TWBKMODU

This package allows users to create or update Web modules, or Web product codes.

P_ModifyPgModule - Page that allows the user to enter/change information in the TWTVMODU table. Uses a call to a javascript procedure to do data validation before the form is submitted.

return_code	<p>Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.</p> <p>N - prints the Info Text message for this package that has the label ERROR_NULL.</p> <p>C - prints the Info Text message with the label ERROR_NOCHANGE.</p> <p>D - prints the Info Text message with the label ERROR_DUPLICATE.</p>
return_msg	A value placed in the Info Text message for ERROR_NULL.
trans_type	Transaction type indicator; U - update, C - create
*_in	Input parameters - will be null if successful. If not null, the update or insert failed, and the user must resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed.

P_OptionPgModule - Page that gives user option of creating or updating an existing module.

return_code	<p>Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.</p> <p>Valid values are: S - Success, N - Error, Record Not Found, E - Other Error, D - Successful Delete</p>
-------------	--

P_SaveModule - Updates or inserts a row into the TWTVMODU table, depending on user inputs and the transaction type.

trans_type	Transaction type indicator. Valid values: U - update, C - create.
*_in	Fields from the data entry form that correspond to the database tables.

TWBKRETC

This package permits users to update Login Return Locations.

Return Locations allow the institution to direct the user's session after login, depending on where the user was when the session timed out or otherwise became not valid. Login Return Locations may be set based on the PL/SQL package name being executed or the Module associated with the package.procedure being executed at the point that the user is forced to login.

Procedures

P_ModifyPgReturnLoc - Page that allows the user to enter or change information in the TWGBRETC/TWGRRETC tables. Uses a call to a javascript procedure to do data validation before the form is submitted.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
return_msg	A message to be displayed on screen.
trans_type	Transaction type indicator. Valid values: U - update, C - create.
*_in	Input parameters - will be null if successful. If not null, the update or insert failed, and the user must resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed.

P_OptionPgReturnLoc - Page that gives user option of creating or updating an existing Return Location.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
-------------	---

`P_SaveReturnLoc` - Updates or inserts a rows into the TWGBRETC/TWGRRETC tables, depending on user inputs and the transaction type.

<code>trans_type</code>	Transaction type indicator. Valid values: U - update, C - create.
<code>*_in</code>	Fields from the data entry form that correspond to the database tables.

TWBKSERR

This package displays any errors raised by the application server and any unhandled exceptions. The text of the error messages is delivered as Info Text - you can customize it for your institution as you would any other delivered Info Text.

You will need to configure your DAD to point to TWBKSEERR. For more information on configuring your DAD, please refer to the *Banner Middle Tier Implementation Guide*.

TWBKSLIB

This package contains the Web Tailor procedures that you can override with local code.

It interacts with the new `GB_THIRD_PARTY_ACCESS` API that is called whenever Web Tailor inserts or updates data on the GOBTPAC table.

TWBKUROL

This package allows users to update Self-Service user roles (e.g., Web Tailor Administrator).

`P_SaveUserRoles` - Updates or inserts a row into the TWGRROLE table, depending on user inputs and the transaction type.

<code>*_in</code>	Fields from the data entry form that correspond to the database tables.
-------------------	---

`P_SelectUser` - Page that gives user option of creating or updating a user's roles.

<code>return_code</code>	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
--------------------------	---

`P_UpdateUserRoles` - Page that allows the user to enter or change user role information.

user_id	ID of user to be update.
---------	--------------------------

TWBKVALD

This package contains functions and procedures which perform data validation activities. Includes PL/SQL code which generates a javascript client-side forms validation function, and general functions which check user input on the back-end.

Functions

`F_JavaScriptValidate` - Prints the javascript function that validates the form field names in `field_tab`.

The following functions are used to check certain types of fields for errors.

`F_DateSeparatorNotFound` - It returns TRUE when date separator / is not being found in 3rd and 6th positions of date field. The correct date format is MM/DD/YYYY.

`F_ErrorMonthDay` - Returns:

- string MONTH if there is an error in MM part of MM/DD/YYYY format
- string DAY if there is an error in DD part of MM/DD/YYYY format,
- string NUMERIC if a character was entered into a Date field,
- string NOTLONG if the string entered in a date field is not long enough for MM/DD/YYYY format.

`F_CharInNumericField` - It returns TRUE if a character was entered in a numeric field.

`F_CharInYear` - It returns TRUE if a character was entered in YYYY part of date format MM/DD/YYYY.

Procedures

`P_AddValidateField` - Adds the specified field to the list of fields to check for null values.

This procedure adds the passed information about a form field to the `val_tab` PL/SQL table. The format of a `val_tab` row is:

```
field_type|check_null|check_numeric|max_length field_name field_desc
```

Where the possible values for the first four fields are:

field_type	Valid values: S - Select list, T - Text field or text area.
check_null	Will be Y by default - N for text fields that are not required but checked for other rules.

check_format	<p>Y=I=Integer, C=Currency, A=Alpha, O=Alphanumeric, N=No check</p> <p>Integer checks that all characters are in (0-9),</p> <p>Currency allows for the addition of one period ('.')</p> <p>Alpha allows (a-z, A-Z)</p> <p>Alphanumeric allows (0-9, a-z, A-Z)</p>
max_length	<p>X by default - this means that no check is performed. If value is an integer, the length of the field may be no longer than the specified value.</p>
max_value	<p>X by default - this means that no check is performed.</p> <p>If value is an integer, the value of the field may be no greater than the specified value.</p>
min_value	<p>X by default - this means that no check is performed.</p> <p>If value is an integer, the value of the field may be no less than the specified value.</p>
The other parameters are:	
field_name	<p>Name of field on the form: <INPUT NAME="field_name"...</p>
field_type	<p>Type of field to be checked. Valid values are:</p> <p>T - Text field</p> <p>S - Select box (A None/NULL value <i>must</i> be first in the <option> list for this select box).</p>
field_desc	<p>Description of field, to be included in error message if the field is found to be null.</p>
<p>P_ValidateFormOpen - Prints an http.formOpen, with one fixed attribute: a javascript call to a function to be performed at the onSubmit event.</p>	
curl	<p>URL of the CGI script.</p>
ctarget	<p>The frame to load the results of the form submit into.</p>
cenctype	<p>The type of encoding used.</p>

<code>cattributes</code>	Any additional attributes to be added to the form tag.
<p><code>P_ValidateResetButton</code> - Prints a form reset button. This procedure is used to standardize the text label on validation/data entry forms.</p> <p><code>P_ValidateSubmitButton</code> - Prints a submit button that validates the data in the form with a JavaScript <code>onClick</code> method.</p>	
<code>validate_tab</code>	PL/SQL table of form fields that need to be checked for not null values.
<code>num_validate</code>	Number of fields to be validated.
<code>button_text</code>	Will override the default text shown on the button (default is currently Save Changes).
<code>cname</code>	Name associated with this button.

TWBKWBIS

This package provides the security and menu processes.

Note: All the PIN-related changes to GOBTAC records are performed by the `GB_THIRD_PARTY_ACCESS` API as of Release 7.0.

Variables

```
minutesPerDay constant number := 1440;
```

Colors used in Web Tailor - borders around update and create boxes

- `create_dk` VARCHAR2(9) -- Darker color in Create box borders
- `create_lt` VARCHAR2(9) -- Lighter color in Create box borders
- `update_dk` VARCHAR2(9) -- Darker color in Update box borders
- `update_lt` VARCHAR2(9) -- Lighter color in Update box borders
- `bottom_right_on` VARCHAR2(1), Y -- Indicates whether or not to print right and bottom borders of Create and Update boxes in Web Tailor administrative pages

Colors used in data-entry forms:

- `header_col` VARCHAR2(9), #cccccc -- Color of headers in some data entry forms
- `lineitem_col` VARCHAR2(9), #ffffff -- Color of line items in some data entry forms.

Variables which keep track of whether or not the HTTP header has been opened and closed, updated by `F_ValidUser`:

- `http_header_closed` BOOLEAN DEFAULT FALSE

- `http_header_opened` BOOLEAN DEFAULT FALSE

Functions

`DISPLAY_FGAC_USAGE` - Contains the values Y or N to indicate whether or not to display the **FGAC Policy exists for** message.

`DISPLAY_VPDI_CODE` - Displays the VPDI code.

NONE	Do not display anything.
CODE	Display only the code.
DESC	Display only the description.
BOTH	Display both the code and description.

`F_CgiBin` - Returns the directory path of the CGI-BIN directory

`F_CheckPositiveInd` - To be used in check box creation in the Web Tailor Administration procedures. If `cursorvalue` in ('A', 'Y'), the function returns CHECKED.

`f_css_link` - Gets the CSS URL defined for the Web page. If it does not exist, the function gets the CSS URL defined for the module. If that does not exist either, the function get the CSS URL defined for the system.

`F_DispInfo` - Prints HTML tags that print an Info Text message and image in table format.

<code>name</code>	The high-level page name, from <code>TWBKWMNU_NAME</code>
<code>label</code>	Message label, from <code>TWBRINFO_LABEL</code> (specifies which piece of text to display, if more than one exists for this page).
<code>msg_text</code>	The message text to be printed. Should be NULL if the text is being fetched from the <code>TWGRINFO</code> table.
<code>msg_type</code>	Indicates which image to print with the message. The default is the standard <code>NOTE</code> type. (Note image printed before text).
<code>value1 AND value2</code>	These are parameters to be inserted into the retrieved text. To be inserted, the text in the database must contain parameter <code>PLACEHOLDERS</code> , <code>< **VALUE1** ></code> or <code>< **VALUE2** ></code> , respectively. These are to be used if user-specific data needs to be displayed as part of a standard error or return message.

`F_FetchWTPParam` - Fetches the parameter value associated with the specified parameter name in the `TWGBPARAM` table.

F_FmtDate - Function to convert a string to a date based on `gubinst_date_default_format`.

<code>str</code>	Character string date variable
------------------	--------------------------------

F_formSelectOption - Function that replaces `htf.formSelectOption`.

<code>cdispvalue</code>	Select list displayed value
<code>cpassvalue</code>	Select list passed value
<code>cselected</code>	SELECTED argument
<code>cattributes</code>	Generic attribute argument
<code><option value=</code>	" <code>cpassvalue</code> " SELECTED> <code>cdispvalue</code>

F_GenRandId - Function to generate a random ID.

<code>id</code>	An ID that will be used to generate a random ID with the same length.
-----------------	---

F_GetParam - Function to return the parameter value for a PIDM and parameter name.

<code>pidm</code>	Banner PIDM
<code>name</code>	Parameter name
<code>encr_flag</code>	Y to decrypt to value before returning

F_Nvl - A customized version of the `nvl` function, it uses a standard uncommon string for replacing the value if it is found to be `NULL`.

<code>check_value</code>	The string to check (<code>NULL</code> or <code>NOT NULL</code>)
--------------------------	--

F_tableDataOpen - Function to generate an HTML `<TD>` tag without the closing `</TD>`.

<code>cvalue</code>	Character value to be displayed inside the data cell
<code>cdp</code>	Value to be used in conjunction with the <code>COLSPEC</code> attribute to set decimal point alignment in a table column. This relates to the <code>DP</code> attribute of the <code><TD></code> tag.
<code>cnowrap</code>	Wrap attribute
<code>crowspan</code>	Row span attribute
<code>ccolspan</code>	Column span attribute

cattributes	Any browser supported <TD attribute> can be supplied here
<p>F_ValidLink - Function to determine if a link name is enabled and valid for current user. It must be called after the call to F_ValidUser.</p>	
name	package.procedure
<p>F_ValidRole - Function to determine if a page is valid for the current user roles. It must be called after the call to F_ValidUser.</p>	
name	Package.procedure or frameset name
name_type	WMNU if a Web menu or procedure, FRAM if a frameset
<p>F_ValidUser - Function to determine if a user is valid and if the user's Web session has timed out. As long as insecure_allowed is FALSE, the procedure will fetch the session information from the SESSID HTTP cookie and perform authentication checks. The cookie will be reset and session information in twbkwses will be updated. If authentication fails, the user will be directed to the login screen.</p>	
pidm	Banner PIDM - this is an OUT parameter. The PIDM retrieved from the session cookie will be passed back here
insecure_allowed	<p>If the page calling F_ValidUser does not require successful authentication to allow access, this is set to TRUE.</p> <p>Most pages should keep the default parameter of FALSE.</p>
check_only	If TRUE, will only check the current sessionid cookie, without setting a new one, or updating the Web session table
close_http_header	If TRUE, will close the http header after setting the sessionid cookie
frameset_in	Specified when called from twbkfram.P_Frameset
menu_in	Specified when called from P_GenMenu .
Return	If TRUE = valid user, FALSE = not valid user or session timed out.

Procedures

`P_Admin_Login` - Logs the person in to Banner Self-Service with their Oracle user ID and password.

The URL should be pointing to the DAD used for FGAC and VBS. For example:

http://servername/vbsdad/twbklogn.P_login

where `servername` is the name of your server, and `vbsdada` is the VBS DAD.

`P_ChangePin` - Generates the change PIN HTML form.

<code>msg</code>	Optional code used to retrieve message string
<code>ret_loc</code>	If <code>P_ChangePin</code> is called with a return location specified, go to the return location after the pin has been changed successfully. The default is to return to <code>TWBKWRUL_BACK_URL</code> for the <code>P_ChangePin</code> procedure. If you use the default, or specify a return location that is a menu, the successful return message will be printed.
<code>ret_type</code>	Will be <code>M</code> if the <code>ret_loc</code> is a Web menu from <code>TWGBWMNU</code> . <code>NULL</code> if the <code>ret_loc</code> is not specified, or if it is a procedure (not a menu) from <code>TWGBWMNU</code> .

`P_CloseDoc` - Generates the ending HTML page.

<code>release_no</code>	Release number to print at page bottom
The following parameters were added for use by the Student Admissions module:	
<code>exit_url</code>	URL which overrides <code>P_Logout</code> in exit link
<code>disp_exit_link</code>	Allows calling proc to prevent display of exit link
<code>back_url</code>	URL for back link
<code>dflt_back_link</code>	Back link text if not defined in the database

`P_DisplayChangePin` - Generic procedure to display a change PIN form.

`P_DisplayInfo` - Prints HTML tags that print Info Text and image in table format.

<code>name</code>	The high-level page name, from <code>TWBKWMNU_NAME</code>
<code>label</code>	Message label, from <code>TWBRINFO_LABEL</code> (specifies which piece of text to display, if more than one exists for this page)

msg_text	The message text to be printed - should be NULL if the text is being fetched from the TWGRINFO tabl
msg_type	Indicates which image to print with the message. The default is the standard NOTE type. (Note image printed before text.)
value1 AND value2	These are parameters to be inserted into the retrieved text. To be inserted, the text in the database <i>must</i> contain parameter placeholders, <***VALUE1**> and <***VALUE2**>, respectively. Use them when user-specific data must be displayed as part of a standard error or return message.

P_FetchSecureCookie - Fetches the value of the security cookie. First checks the standalone security cookie. If not found, tries to find the name of the CP global cookie, and fetch that cookie. Cookie_value will be equal to (webid || pidm) if a cookie is found.

P_FirstMenu - Procedure to generate first menu after logging in/accepting terms of usage.

pidm	Banner PIDM
webid	Dynamically generated Web session ID
usage_btn	String containing text on the button pressed from usage page

P_formSelectOption - Procedure that replaces http.formSelectOption.

cdispvalue	Select list displayed value
cpassvalue	Select list passed value
cselected	SELECTED argument
cattributes	Generic attribute argument
<option value="cpassvalue" SELECTED> cdispvalue	

P_Logout - Procedure to logout of the Web session.

pidm	Banner PIDM, only specified when the Exit button is selected on P_UsagePage. Otherwise, the PIDM will be retrieved from SESSID cookie
------	--

P_MetaForward - Prints an HTML page with META tag headers that forward the browser immediately to ret_location.

P_OpenDoc - Generates the beginning HTML page. It will refer to the Luminis CSS if it is present.

name	Package.procedure name
title_text	Optional text to be appended to the title
header_text	Optional text to be appended to the header
refresh_delay	Number of seconds to stay on current page before refreshing
refresh_url	Location of page to load after the delay
reload_navbar	TRUE if the navbar should be reloaded on loading of this information page.
additional_body_onload	Any additional text to be used with the JavaScript onLoad event handler within the html BODY tag
exit_url	URL which overrides P_Logout in exit link
disp_exit_link	Allows calling proc to prevent display of exit link
back_url	URL for back link
dflt_back_link	Back link text if not defined in the database

P_PGenMenu - Procedure to generate the Web menus.

name	Menu name, package.procedure to call
msg	Optional code used to retrieve message string
exit_url	URL which overrides P_Logout in exit link.
message_type	Determines how P_printmessage outputs the message. Valid values are: 1 = Error 2 = Warning 3 = Note The default value is NULL.

P_ProcChangePin - Procedure to process the Change PIN page.

oldpin	Old PIN number.
newpin	New PIN number.
verpin	Verification of new PIN number.
ret_loc	If P_ChangePin was called with a return location specified, use the return location after the PIN has been changed successfully. The default is to return to TWBKWRUL_BACK_URL

for the `P_ChangePin` procedure. If you use the default, or specify a return location that is a menu, the successful return message will be printed.

<code>ret_type</code>	Will be <code>M</code> if the <code>ret_loc</code> is a Web menu from <code>twbkwmenu</code> . <code>NULL</code> if the <code>ret_loc</code> is not specified, or if it is a procedure (not a menu) from <code>twbkwmenu</code>
-----------------------	---

`P_RedirectURL` - Sends the user to the specified URL, using either `P_MetaForward` or `owa_util.redirect_url`, depending on the value of `twbkwurl_http_forward_ind`.

`P_Securelogin` - Used for administrative users who need a secure login

`P_SecurityAnswer` - Procedure to generate the Security Question and Answer HTML form.

<code>msg</code>	Optional code used to retrieve message string
<code>sid</code>	User ID

`P_SecurityQuestion` - Procedure to generate the Security Question and Answer HTML form.

<code>msg</code>	Optional code used to retrieve message string
<code>ret_loc</code>	If <code>P_SecurityQuestion</code> is called with a return location specified, go to the return location after the pin has been changed successfully. The default is to return to <code>TWGBWRUL_BACK_URL</code> for the <code>P_SecurityQuestion</code> procedure. If you use the default, or specify a return location that is a menu, the successful return message will be printed.
<code>ret_type</code>	It will be <code>M</code> if the <code>ret_loc</code> is a Web menu from <code>TWGBWMNU</code> ; it will be <code>NULL</code> if the <code>ret_loc</code> is not specified, or if it is a procedure (not a menu) from <code>TWGBWMNU</code> .

`P_SetParam` - Procedure to store the value of a parameter in the `GORWPRM` table.

<code>pidm</code>	Banner PIDM
<code>name</code>	Parameter name
<code>val</code>	Parameter value
<code>encr_flag</code>	<code>Y</code> to encrypt the value before saving. If the value is being encrypted, it must not be longer than 21 bytes.

`P_tableDataClose` - Procedure to generate a `</TD>` tag.

P_tableDataOpen - Procedure to generate an HTML <TD> tag without the closing </TD>.

cvalue	Character value to be displayed inside the data cell
cdp	Value to be used in conjunction with the COLSPEC attribute to set decimal point alignment in a table column. This relates to the DP attribute of the <TD> tag
cnowrap	Wrap attribute
crowspan	Row span attribute
ccolspan	Column span attribute
cattributes	Any browser supported <TD attribute> can be supplied here

P_ValidateChangePin - Generic procedure to validate a PIN change and update the database.

pidm	Banner PIDM
oldpin	Old PIN number
newpin	New PIN number
verpin	Verification of new PIN number

P_WWWLogin - Procedure to generate the login Web page. It will check to see if TWGBLDAP_PROTOCOL contains LDAP or LDAP_S. If so, it will not display the **Forgot PIN?** button on the login page.

This procedure selects user ID and password literals from the TWBKPARAM table.

TWBKWRUL

The procedures in this package lets you update the Web Rules for your institution.

P_ModifyPgWebRules - Page that allows the user to enter or change information in the TWGBWRUL table. Uses a call to a javascript procedure to validate the data before the form is submitted.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
return_msg	A message to be displayed on screen

*_in	Input parameters - if null, the update or insert was successful. Otherwise, the user is asked to resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed.
P_SaveWebRules - Updates or inserts a row into the TWGBRUL table, depending on user inputs and the transaction type.	
trans_type	Transaction type indicator. Valid values: U for update, C for create
*_in	Fields from the data entry form that correspond to the database tables

TWBKWMNU

This package creates pages that allow users to add and update Web menus or procedures.

P_ModifyPgWebMain - Page that allows the user to enter or change information in the TWGBWMNU table. Uses a call to a javascript procedure to validate the data before the form is submitted.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
return_msg	A message to be displayed on screen.
trans_type	Transaction type indicator. Valid values: U for update, C for create.
*_in	Input parameters. Will be null if the insert was successful, otherwise, the user must resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed.

P_OptionPgWebMain - Page that gives user option of creating or updating an existing Main Text page.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure.
-------------	---

P_SaveWebMain - Updates or inserts a row into the TWGBWMNU table, depending on user inputs and the transaction type.

trans_type	Transaction type indicator. Valid values: U for update, C for create.
*_in	Fields from the data entry form that correspond to the database tables.

TWBKWINF

This package contains procedures that allow users to update Info Text and error messages.

P_CopyToLocal - This procedure will copy all baseline Info Text rows to local Info Text rows. If there are any local rows, it won't overwrite them. If a parent menu record is needed (due to the new source_ind column) then create a parent row first.

P_ModifyPgInfoText - Page that allows the user to enter or change information in the TWGRINFO table. Uses a call to a javascript procedure to do data validation before the form is submitted.

return_code	Indicates that the procedure has been called from another procedure, and that some message should be displayed to explain the outcome of the calling procedure
return_msg	A message to be displayed on screen
trans_type	Transaction type indicator. Valid values: U for update, C for create
*_in	Input parameters. Will be null if the update or insert failed, otherwise, they will not be null, and the user will be asked to resubmit the data. The values the user entered into the form will be passed back into this procedure and displayed

P_OptionPgInfoText - Page that gives user option of creating or updating an existing set of Info Text entries (repeating records).

P_ReorderInfoText - Back-end process which reorders the Info Text for the specified menu, then returns to the Reorder Info Text page.

name_in	Name of the menu, from TWBRINFO_NAME
orig_seq_tab	Table of original sequence numbers as printed on the Reorder Info Text page
new_seq_tab	Table containing the values of Info Text sequence number pull-down lists on the page

label_tab	Table containing the labels as printed on the Reorder Info Text page
P_ReorderPgInfoText - Page that displays all current Info Text, gives links to update each item, and an interface for reordering elements.	
name_in	Menu name from TWBRINFO_NAME
return_msg	Return message to be printed
return_code	Error return code
P_SaveInfoText - Updates or inserts a row into the TWGRINFO table, depending on user inputs and the transaction type.	
trans_type	Transaction type indicator. Valid values: U for update, C for create
*_in	Fields from the data entry form that correspond to the database tables.

Package specs for Luminis Channels for Banner

The package specs for Luminis Channels for Banner are delivered with Web Tailor (the package bodies are delivered with Luminis Channels for Banner). They are described in this section.

twbccomn.sql

This is the package spec for channel functions and procedures. It also stores the common methods which are used in channel processing. The package body is delivered with Luminis Channels with Banner.

twbcmain.sql

This is the package spec for the main channel package. Luminis classes call it to retrieve the XML used to render the channels in Luminis. The package body is delivered with Luminis Channels with Banner.

twbcmxml.sql

Individual channel packages will use methods from this helper package to generate XML using the Oracle XDK. XML is used in Luminis to render channels.

Graphic Elements

This chapter presents instructions on replacing or modifying the images delivered as part of Web Tailor and the Banner™ Self-Service products.

Review and customize graphic elements

Graphic elements are images that can be customized to display at various places in Self-Service Banner. They can be placed next to menu items, error and warning messages, links, Info Text, and the like.

About this task

You will want to review the delivered graphic elements to make sure they are appropriate for your institution.

You can use Web Tailor to customize the graphics and icons that appear on your Web pages, or to define new ones. To customize the images that display, perform the following steps:

Procedure

1. Select **Graphic Elements** from the Web Tailor Administration Menu.
2. Select the **Create** button to create a new element, or select an element from the pull-down menu.
3. Enter the name of the element. If you are customizing a delivered element, you may want to rename the element to something unique to your institution.
4. Enter the URL that points to the element on the web server.
5. Enter the image's width and height.
6. Enter any alternate text to be processed by a user agent such as a screen reader. This will help a visually-impaired user understand how the graphic element is used.

Delivered images

Delivered images can be found in the following locations.

- Faculty Self-Service - images are delivered in `/facweb/gifs` and deployed in `/facgifs` on the web server to support the Faculty Self-Service product.
- Student Self-Service - images are delivered in `/stuweb/gifs` and deployed in `/stugifs` on the web server to support the Student Self-Service product.
- Web Tailor - images are delivered in `/wtlweb/gifs` and deployed in `/wtlgifs` on the web server to support the Web Tailor application.

Tag Mapping

This appendix has been provided to help you customize the delivered Web pages. It identifies which items on the pages are created and influenced by which tags.

There are more style classes related to tables than are shown here. For example, there are style classes for DATAENTRY, DATADISPLAY, and PLAIN type tables and their components (table headers and table cells).

Style rules for form controls are defined for each HTML form control element. There is a single style rule for each of INPUT, TEXTAREA, and SELECT.

Error and warning messages use the errortext and warningtext classes respectively.

Note: The text SPANS and other tag classes describe the objects; the DIV classes specify where they appear on the web page.

This appendix covers three different web pages.

Login page

This is the Login page.

Name	Description
BODY	The entire screen.
H1	The level 1 header.
pageheaderlinks	The region directly under the level 1 header. Usually there would be links to other pages, but on this page there aren't any.
TABLE.infotexttable	This class defines styles for the table that will be populated with Info Text.
TD.indefault1	This class defines styles for the table's data cells.
infotext	This class defines text styles for Info Text.
TABLE.dataentrytable	This class defines styles for the table that contains a form with data entry fields.
TD.delabel	This class defines styles for table data cells containing labels.
fieldlabeltext	This class defines text styles for labels.
TD.dedefault	This class defines default styles for table data cells.

Name	Description
releasetext	This class defines styles for release text.
headerwrapperdiv	The DIVs identified by the pageheaderdiv1 and pageheaderdiv2 classes.
pageheaderdiv1	The DIV class that contains H1, which contains the page header.
pagetitlediv	The DIV class that contains H2, which contains the page title (not applicable in this case).
pagebodydiv	The DIV class that contains all the page-specific content. In this case, the Info Text, data entry fields, and buttons.
infotextdiv	The DIV class that contains the Info Text.
pagefooterdiv	The DIV class that contains footer information for this screen.
poweredbydiv	The place where the Powered by Ellucian graphic is displayed. The Powered by Ellucian graphic is no longer used.

Web Tailor menu

This is an example of a menu page.

Name	Description
BODY	The entire screen.
H1	The level 1 header.
pageheaderlinks	Links to other pages, e.g., Personal Information, Student Services & Financial Aid, etc.
H2	The level 2 header, which contains the page title. In this case it says Web Tailor Menu.
TABLE.menuplaintable	This class defines styles for a table that has menu links.
TD.mpdefault	This class defines styles for table data.
releasetext	This class defines styles for release text.
headerwrapperdiv	The DIVs identified by the pageheaderdiv1 and pageheaderdiv2 classes.

Name	Description
pageheaderdiv1	The DIV class that contains H1, which contains the page header.
headerlinksdiv	The DIV class that contains the links to other web pages. In this case the links are Personal Information, Student Services & Financial Aid, etc.
pagetitlediv	The DIV class that contains H2, which contains the page title (in this case, Customize the selected Module).
pagebodydiv	The DIV class that contains all the page-specific content.
pagefooterdiv	The DIV class that contains footer information for this screen. In this case, it contains Release 5.2.
poweredbydiv	The place where the Powered by graphic is displayed. The Powered by Ellucian graphic is no longer used.
headevlinksdiv2	The DIV class that contains search box 2.
submenulinktext2	This defines menu links.
staticheaders	The DIV class that contains static headers.

Customize the selected module

This is an example of an application page.

Name	Description
BODY	The entire screen.
H1	The level 1 header.
pageheaderlinks	Links to other pages, e.g., Personal Information, Student Services & Financial Aid, etc.
H2	The level 2 header, which contains the page title. In this case it says Customize the selected Module.
infotext	This class defines styles for Info Text.

Name	Description
fieldrequiredtext	This class defines styles for the graphic that indicates that a field is required.
requiredmsgtext	This tag defines styles for the required field message text. In this case, indicates a required field.
TABLE.dataentrytable	This class defines styles for the table that contains a form with data entry fields.
TD.delabel	This class defines styles for table data cells containing labels.
TD.dedefault	This class defines default styles for table data cells.
fieldlabeltext	This class defines text styles for labels.
backlinktext	This class defines styles for a back link, in this case it says <code>Select another Module to customize.</code>
headwrapperdiv	The DIV class that contains BODY, H1, and pageheaderlinks.
pageheaderdiv1	The DIV class that contains H1, which contains the page header.
headerlinksdiv	The DIV class that contains the links to other web pages. In this case, <code>Personal Information, Student Services & Financial Aid, etc.</code>
pagetitlediv	The DIV class that contains H2, which contains the page title (in this case, <code>Customize the selected Module</code>).
pagebodydiv	The DIV class that contains all the page-specific content. In this case, the Info Text, data entry fields, pull-down lists and buttons.
infotextdiv	The DIV class that contains the Info Text.
backlinkdiv	The DIV class that contains the Back link and text.
headlinksdiv2	The DIV class that contains the search box.
staticheaders	The DIV class that contains static headers.