ellucian™

Banner Web Tailor
User Guide

Release 8.6.1
March 2014

Revision History

Publication Date	Summary
March 2014	New version that supports Banner Web Tailor 8.6.1 software.

Contents

Web Tailor Overview

This chapter provides a summary of the functions of Banner™ Web Tailor, and specifies how users access the system. It also includes a description of features that are available for all the Banner Self-Service applications and are set up in Web Tailor.

Functions

Banner Web Tailor lets you build the look, feel, and unique personality of all your institution's web applications, so you can personalize your institution's interface to the world. Web Tailor delivers customizable global web rule definitions and procedures, customizable menus, menu items, graphics and text definitions.

You must install Web Tailor before installing any other Banner Self-Service application. For more information, refer to the *Banner Middle Tier Implementation Guide*.

The Web Tailor functions are available from the Web Tailor Menu, except for **Change Security Question** and **Change your PIN**, which are found on the Personal Information menu. The Web Tailor functions will be described in detail later in this guide.

Web menus and procedures

This function allows you to define the menus that will appear on your institution's web pages for the different Self-Service applications, and specify the procedures behind them. For more information, see "Web menus and procedures" on page 45.

Menu items

This function allows you to define the items that will appear on the menus on your institution's web pages. For more information, see "Menu items" on page 49.

Information Text

This function allows you to add or customize Information Text (Info Text). Info Text can be:

- Instructions on how to use a page

- Help for the page

- Error messages

    For more information, see "Info Text" on page 52.

User roles

This function allows you to change the role or roles that a person has been assigned to. For example, you can select the check boxes for Advancement Data Tailor and Development Officer to give someone those roles. For more information, see "User roles" on page 55.

Web rules

This function allows you to define certain rules for your institution's web pages. For example, you can identify the number of minutes a person can be inactive before they are timed out, or specify the format for the date and time information that appears on your pages. For more information, see "Web rules" on page 56.

Web modules

This function allows you to modify a specific Self-Service application, e.g., Accounts Receivable, Student Self-Service, Web Tailor Administration, etc. You can make specific changes to that one application.

For example, you can enter a location URL of the Cascading Style Sheet (CSS) that applies to Advancement Self-Service alone. The Advancement Self-Service application would use that CSS and the other applications at your institution could use different ones. Each of your modules could have a different look-and-feel. For more information, see "Web modules" on page 62.

Global user interface settings

This function allows you to set up rules that will apply to your institution's web pages as a whole. You can specify:

- Header information

- The location URL of CSSs that control the pages' look-and-feel

- The location URL of CSSs that control the look-and-feel of your Help text

- The location URL of where your Help text files are stored

- Images that represent errors and warnings

- An image that indicates that a field is required

  For more information, see "Global user interface settings" on page 63.

Web Tailor parameters

This function allows you to customize parameters used in Web Tailor processing, such as the maximum length of PINs. You must exercise great care when modifying these parameters. For more information, see "Web Tailor parameters" on page 65.

Graphic elements

This function allows you to specify the images that will be available to be used on your web pages. For each image, you can specify its name, the directory where it is

located, and its height and width. For more information, see "Graphic elements" on page 74.

Login return location

Use this function to specify the page you would like to be displayed when a user is timed out, then logs back in. For more information, see "Login return location" on page 75.

Web Tailor overrides

This page allows you to replace certain procedures and functions with your own under certain circumstances. This is necessary because you may have a stand-alone product you would like to use with the Self-Service products, and you need to use some of the procedures and functions in the other system. If an override is defined, that code will be run instead of the Web Tailor code. For more information, see "Web Tailor overrides" on page 76.

Advancement Self-Service rules

This function, although stored in Web Tailor, is used only by those who want to customize Banner Advancement Self-Service. For information on how to do this, please refer to the *Banner Advancement Self-Service User Guide.*

You must have the Advancement Data Tailor role assigned to your user ID to see this menu item.

Finance site preferences

This function, although stored in Web Tailor, is used only by those who want to customize Finance Self-Service. For information on how to do this, please refer to the *Banner Finance Self-Service User Guide.*

You must have the Finance Data Tailor role assigned to your user ID to see this menu item.

LDAP administration

This function allows you to override the settings on the Enterprise PIN Preferences Form (GUAPPRF) in Banner General and use an LDAP server to authenticate user logons instead of the Web Tailor logic. For more information, see "LDAP administration" on page 77.

Channel maintenance and channel preference

These items will appear on your menu only if you are using Web Tailor with Luminis® Channels for Banner and Luminis. For more information, see the *Banner Middle Tier Implementation Guide.*

Changing the security question

This function allows your end users to change the security questions and answers that they can use to access their account if they forget their PINs. See the *Banner Web General User Guide*.

Changing a PIN

This function enables users to update their PINs. See the *Banner Web General User Guide*.

Accessing Web Tailor

The Self-Service applications provide two types of web access, non-secured and secured. Users can access non-secured areas without needing to enter a user ID and PIN.

> **Note:** Web Tailor requires secured access so you must log on with an ID an PIN to access it.

To access Web Tailor, perform the following steps:

1. Select **Secure Area** from the home page.

2. Enter your ID and PIN on the logon page.

   > **Note:** The PIN is case-sensitive. Depending on your institution's settings, the User ID may also be case-sensitive.

3. Select **Web Tailor Administration**.

Technical Basics

This section describes basic technical information about Banner™ Web Tailor. The following topics are covered:

Technical configuration

Banner Self-Service uses PL/SQL stored program units, which streamlines the call and retrieval process for query and update functionality.

The Self-Service programs are written in PL/SQL and are stored as PL/SQL packages, procedures, and functions. The Self-Service applications are designed around a library of stored subprograms.

The architecture eliminates SQL parsing for each PL/SQL statement by combining PL/SQL stored program units with pre-compiled Oracle subprograms stored in the Oracle database. This design substantially reduces the call processing overhead required for the Self-Service applications to communicate with Oracle. It also promotes memory sharing for multiple users executing procedures and functions within the same package.

Data flow process

The Self-Service applications share a common data flow, which consists of the following steps that exchange information between a web client and the Oracle host:

1. The web client browser software accesses a web page or completes an HTML form that is sent to the institution's web server.

2. The institution's web server software interprets the URL (Uniform Resource Locator) and identifies it as either a CGI-compliant (Common Gateway Interface-compliant) program or as a procedure to access static HTML pages. If it is a CGI-compliant program, the web server software invokes the Oracle Application Server.

3. The Oracle Apache PL/SQL Agent software reads and decodes the passed data, connects to the Oracle database via SQL/Net and executes a remote Oracle stored procedure that resides on the Oracle database.

4. The stored procedure executes other stored procedures and functions to perform SQL operations and dynamically generate HTML directives that will be passed back to the `mod_plsql` component of the Oracle HTTP server.

5. The `mod_plsql` component of the Oracle HTTP server passes the data back as reformatted HTML tags to the institution's web server software that communicates with the web client.

Accessing the Self-Service applications

Two types of access are provided in the Self-Service applications:

- *Non-secured access* - does not require users to enter an ID and PIN to log on to the homepage, providing your institution with a way to effectively market and publicize on the web. Typically, institutions display the **Prospective Students**, **Apply for Admission**, **Campus Directory**, **General Financial Aid**, **Class Schedule**, **Course Catalog** and **Alumni and Friends** links here. The home page displays several icons for Public Access (non-secured) information.

  Note: Part of the admissions process does require the user to create an ID and select a PIN, which will be required if the user needs to enter more application information at another time or if they want to view the status of their application after they submit it.

- *Secured access* - requires users to enter a valid ID and PIN to log on, and gives them access to information from Banner. The home page displays **Secure Area**. After the user enters a valid ID and PIN, the Main Menu display includes the items that are valid for the user's web roles.

  For details about PINs, please refer to Chapter 4, "PINs". In addition, there is more information about web session processing and security in Chapter 5, "Technical Reference".

Security and the browser's back button

Because Web Tailor assures security, the **Back** button can be used on many browsers.

Of course, the only really safe way to prevent other users from logging in after a person leaves a computer terminal is to log out and close the browser. That is the best protection against unauthorized access to personal information. If users merely cleared the disk and memory cache, it would not erase the security cookie, which would allow another user access.

Interaction with APIs

Web Tailor supports the integration of Banner and Banner Self-Service with Application Programming Interfaces (APIs).

An API is a program comprised of three different packages:

1. The primary package that contains the create, update, and delete functions

2. A rules package that contains subprograms that support the primary package

3. A strings package that contains error messages and functions

Programs within Banner (including forms, C processes, and the Self-Service packages) call common functions and procedures in the APIs as needed. External systems that create, update, or delete information stored in Banner tables can also call the APIs.

Web Tailor calls the Third Party Access API (`GB_THIRD_PARTY_ACCESS`) for all insert and update processing for the GOBTPAC table.

Any database rollbacks or commits have been changed to call the new `gb_common.p_rollback` or `gb_common.p_commit` procedures. This ensures that the appropriate actions are taken to publish or discard synchronization messages.

API error handling for all the Self-Service products is done via the TWBKWBIS package. The TWBKSERR package handles all unhandled exceptions.

User interface

The look and feel of the Banner Self-Service products are defined by the selection of one of two distinct themes, default and cascade. The use of themes allows the same content to be displayed in different ways. For more information on the available themes, please see:

- "Default theme" on page 19

- "Cascade theme" on page 21

To select a theme, perform the following steps:

1. Click **WebTailor Administration**.

2. Click **WebTailor Parameters**.

3. Click **UI_THEME**.

4. Enter the parameter value for the theme you would like to select. Available themes include the following:

   - `cascade` - displays the Cascade theme.

   - `default` - displays the Default theme.

5. Click **Submit Changes**.

Default theme

The default theme has the following design elements.

- Use the Verdana font

- Includes separate bars consistent with the Banner 7 and 8, Luminis 5look and feel.

- Provide a consistent place to render error messages to assist the user in determining when validation issues occur.

It is recommended that you set the font size in your browser to `Medium` (View>Text Size>Medium). If you set the size to `Largest`, a small line will appear to the right of each tab.

> **Note:** In the Secure Area, the options on the Main Menu page (twbkwbis.P_GenMenu) are also displayed across the top of the page as tabs. (There are some exceptions to this, however, such as the WebCT menu item.)

You can choose a menu item or a tab to proceed to the corresponding page. The tab will remain at the top of the page while you access the pages associated with it, and it will be highlighted.

While you are navigating through any of the lower-level menus, that tab will still be highlighted.

There are two exceptions:

1. If you reach a page that is not secured (you can access it without accessing the Secure Area), there are no tabs at the top of the Web page.

2. If you reach a page that is owned by a product that you do not have access to, there will be tabs, but none of the tabs will be highlighted.

   For example, if a user who does not have Web Tailor Administrator access is on the Personal Information menu, the Personal Information tab is highlighted. However, when that user selects **Change Security Question**, the Personal Information tab is no longer highlighted because the Security Question and Answer page is owned by Web Tailor Administration.

Accessibility initiative/enhanced user interface

The User Interface (UI) for Banner Web Tailor has been designed to be understood easily by casual users, and it includes accessibility features.

The goal is to make the content understandable and navigable for any individual who uses adaptive technology such as screen readers and screen enlargers. This includes not only making the language clear and simple, but also providing easy steps to complete a user task and designing mechanisms for navigating within and between pages. Providing navigation features and orientation information in pages maximizes accessibility and usability.

The look-and-feel settings for Web Tailor are stored in Cascading Style Sheets (CSS). By providing more CSS in Web Tailor, you can create and manage the look and feel for your institution.

The benefits are:

- Improved usability for all constituents, including the disabled

- A more flexible user interface with greater ability to customize the look and feel to meet institutional needs

- Greater extensibility for local modifications due to increased use of industry standards, including W3C guidelines

- An improved foundation for future changes in web technology

This includes:

- Text describing graphics appears in the Status Bar and Bubble Help. There are no text-based links under the graphics in the navigation section of the Page Header because this would cause redundancy when using a screen reader.

- Tags are included with graphics so screen readers can identify them.

- High-level menu items are text, not graphics.

- Style sheets are used extensively. They support highly structured documents, which are easier for screen readers to process.

- Tables have been designed to make them easy to read and understand, and they have summaries and captions.

- Information Text (Info Text) has been designed to be easy for screen readers to process.

- Abbreviations and acronyms are defined by tags to be spoken correctly. For example, screen readers would try to pronounce URL as a word if it was not marked correctly.

- Text size can be customized, increasing it from small to large, using the text size features found in the View menu in the browsers.

- Imbedded Skip links allow screen readers to move past repetitive text to the main content of the page.

- Keyboard shortcuts are included for Help, Menu, and Exit:

    - ALT + 2: Exit

- ALT + 3: Menu
- ALT + H: Help

- Text links or anchor links appear at the top of each page to allow users to go directly to the text they want to read.

- `Go To Top` links appear at the bottom of pages to take users directly to the top.

- Form field labels are associated with form fields.

- Users are warned when links will spawn new windows.

- A DTD identifier exists on all web pages.

- A language attribute exists in the HTML tag on all pages.

- Web pages are divided into sections via DIV tags.

- TH tags are used for table headers (rather than TD tags), and they include a scope attribute.

- Device-independent alternatives have been provided for device-dependent javascript (e.g., you can use either the keyboard or a mouse).

Frames

It is recommended that your institution not use HTML frames in new or future UI designs for Self-Service products. The use of frames in HTML design causes problems with accessibility, printing, compatibility, and browsing on mobile devices. For these reasons, frames are not supported in Self-Service.

Cascade theme

The Cascade theme provides Web 2.0 functionality within Banner Self-Service. This theme adds rich user interaction to Banner Self-Service such as auto complete functionality, button mouse-over state changes and in page refreshes to provide the end user a compelling user experience.

Note: Javascript must be enabled for the Cascade theme to be used.

The Cascade Theme introduces some new internet standard design concepts to Banner Self-Service.

- The page hierarchy is now provided on a page by page basis through the use of a breadcrumb trail. A breadcrumb trail is a popular way to give end-users a way to keep track of their locations within Banner Self-Service. For more information, please see "Breadcrumb validation" on page 103.

- A back button is provided next to the breadcrumb trail that uses the Back Link URL. This relationship is defined in Web Tailor through use of the Back Link URL field. If no Back Link URL is specified, the back button will not display.

- More granular use of CSS to allow web designers to tailor the look and feel of the theme. This allows Banner Self-Service to have more flexibility to adopt the look and feel of the institutions brand. With this finer granularity, however, there is an increased level of complexity when making these changes.

- With the Cascade Theme applied, Banner Self-Service provides a more robust way in the which the end-user can navigate throughout the system. Through in page menu refreshes the end-user no longer has to go from one page to another, but has the navigation brought into the page they are currently viewing.

- The "Search" feature in Banner Self-Service is more accurately depicted in Banner Self-Service as "Find a Page". The new "Find a Page" functionality allows the user to type a few characters and automatically retrieve those pages that best match without ever having to leave the page.

Main menu

The Cascade theme does not display a Main menu in the secured area. If you have items on the main menu that do not display as tabs, you will need to either create a new main menu tab to display this content or move the items on the main menu that do not display as a tab to an existing tab.

Menu pages

Global Menu Bottom Links do not display on menu pages in the Cascade theme. Global Menu Bottom Links will only display on menu pages that use the Default theme.

Cascade theme accessibility

The Cascade theme contains some items that are not accessible by a screen reader. The Cascade theme has been built with an accessibility mode option for users using a screen reader to access the application. In accessibility mode, the session will run under the default theme which is fully accessible. To enable accessibility mode, for a session, an accessibility URL can be created using the following format:

```
http://<server>:<port>/<dad>/
twbkwbis.P_GenMenu?name=homepage&accessibility=true
```

Naming conventions

The Banner Self-Service products are made up of packages, C programs, HTML files, and GIF and JPG files, javascript, and Cascading Style Sheets (CSS). Each object's name is eight characters long, and the position of each letter identifies a particular quality or attribute of the package or program.

For example, the eight letters in BWGKOGAD identify it as a Web General package in the Overall module that is used for general address processing. The package specification

can be found in the file `genweb/dbprocs/bwgkogad.sql`, while the package body will be in the file `genweb/dbprocs/bwgkoga1.sql`.

Exceptions to these rules are explained on the next pages. Most notably, the packages TWBKLIBS and TWBKWBIS are the Web Tailor Library package and the Web Tailor Basic Support package, respectively, and the HTML file `homepage.htm` is the example homepage. Also, for all but very small packages, look for package bodies in a file with the same first seven characters as the package name, but with a one (1) in the eighth position.

Web Tailor object names

All Web Tailor package names begin with the four characters `TWBK`, and do not have a module code in the fifth position. In a Web Tailor package, the characters in positions 5 through 8 are all devoted to the unique package name. In this respect they follow the same naming convention as the HTML, GIF and JPG files for the last four characters of a Web General object.

Position(s)	Assignment
Positions 1 and 2	The letters `BW` occupy positions 1 and 2 in all Web General package names. The letters `TW` occupy positions 1 and 2 in all Web Tailor package names.
Position 3	Identifies the associated baseline product: A = Advancement Self-Service B = Web Tailor C = Common Web D = Advancement Officers (part of Advancement Self-Service) F = Finance Self-Service (part of Employees Self-Service) G = Web General L = Faculty and Advisor Self-Service P = Employee Self-Service R = Financial Aid Self-Service (part of Student Self-Service) S = Student Self-Service (includes Admissions Self-Service) T = Accounts Receivable (part of Student Self-Service)
Position 4	Identifies object type: K = Oracle database package R = C report A = C application support object H = HTML file G = GIF file or JPG file
Position 5	For packages and C files, identifies module: O = Overall
Positions 6-8	For packages and C files, a unique three-character name
Positions 5-8	For HTML, GIF, and JPG files, a unique four-character name For Web Tailor packages, a unique four-character name

Web database object naming conventions

> **Note:** The Banner Object:Access product uses slightly different standards for its procedures and functions.

Database Object	Naming Convention
Cursors	Up to 29 free-form characters followed by C. (For example, `GetInfoC`)
Functions	F_ followed by up to 28 free-form characters. (For example, `F_ValidUser`).
Procedures	P_ followed by up to 28 free-form characters. (For example, `P_OpenDoc`)

Cascading Style Sheets

Web Tailor uses Cascading Style Sheets (CSS) to control the way documents are presented via browsers and are printed.

Cascading Style Sheet (CSS) specifications are maintained by the World Wide Web Consortium (W3C) to help introduce more consistency and structure into Web pages. Using them allows web developers to create a set of styles then apply them to all their Web pages uniformly. If a developer changes a style on the style sheet, all the pages that use that style are affected.

For more information on CSS, see the W3C's CSS page at `http://www.w3.org/Style/CSS/`

You can easily modify your style sheet file using any text editor in order to change the visual style of Banner Self-Service pages. *Appendix A, Tag Mapping*, has been provided to help you customize the Web pages. It shows how a number of items on the pages are created and influenced by specific tags.

> **Note:** When you modify CSS files, and when modified CSS files are redelivered, the new versions of the files must be copied to web server. See Chapter 2, "Configuring Self-Service Banner," in the *Banner Middle Tier Implementation Guide*, for instructions on deploying files on the web server

The style sheets delivered with the Self-Service products are external style sheets. If you place different styles in the <HEAD> tag of an HTML file, or within the HTML file itself, you will override the default styles.

Using Web Tailor, you can specify a style sheet file location at the following levels:

- *System* - updated via Global User Interface Settings

- *Module* - updated via Web Modules

- *Web page* - updated via Web Menus and Procedures

These file locations are expressed as URLs; for example, `/css/web_defaulthome.css` is the location of one of the style sheets delivered with Web Tailor.

As each page is displayed to the user, the browser formats the page and its contents according to the rules in one of the style sheets (system, module, or Web page). The user-defined style sheet location that has the most specific information for that page is the one that the page uses. It places the location in an HTML `<LINK>` tag inside the `<HEAD>` tag.

Style sheets

Web Tailor CSS files are stored in the `wtlweb/htm` subdirectory.

During Banner implementation, the CSS files must be copied to your web server. See Chapter 2, "Configuring Self-Service Banner," in the *Banner Middle Tier Implementation Guide*, for instructions on setting up CSS files on the web server.

Web Tailor includes the following CSS files for the default theme:

- `web_defaulthome.css`

  This CSS contains the setting for the Self-Service home page.

- `web_defaultmenu.css`

  This CSS contains the settings for the Self-Service menu pages.

- `web_defaultapp.css`

  This CSS contains the settings for the Self-Service application pages.

  > **Note:** `web_defaulthome.css`, `web_defaultmenu.css` and `web_defaultapp.css` are designed to be used together. It is recommended that your institution use them.

- `web_defaulthelp.css`

  This CSS is present at all times, in addition to the CSS defined for the application pages. It defines any additional styles necessary to present Help text to the user.

- `web_defaultprint.css`

  This CSS controls the presentation of printed Web pages. You can modify it if, for example, you want web pages to print without their Info Text.

Web Tailor includes the following CSS files for the Cascade theme:

- app-overrides.css

  This CSS contains styles that override the default theme.

- cascasde.common.css

This CSS contains common styles for the Cascade theme.

- cascade.css

  This CSS contains styles for the Cascade theme.

- cascade-luminis.css

  This CSS contains styles for the Cascade theme when using Luminis.

- common-controls.css

  This CSS contains styles for common controls.

- common-platform.css

  This CSS contains styles for the common platform.

- ie6.css

  This CSS contains styles to render pages in the Internet Explorer 6 browser.

- ie7.css

  This CSS contains styles to render pages in the Internet Explorer 7 browser.

- ie8.css

  This CSS contains styles to render pages in the Internet Explorer 8 browser.

- login.css

  This CSS contains styles for the Cascade theme login page.

- jquery.autocomplete.css

  This CSS is used by the autocomplete control and is used for search functionality.

Home page

The home page is a Web menu rather than a static HTML page. You can customize it for your institution just as you would any other menu.

To change the header information on the home page, choose **Web Menus and Procedures**, then select `homepage` from the pull-down list. You can specify the header text, header graphic, the CSS that controls the home page, etc.

To change the contents of the home page, choose Menu Items, then select `homepage` from the pull-down list. You can specify the links that will appear on the home page, and the order in which they will appear. For example, you may choose to display **Secure Area**, **Campus Directory**, and **Apply for Admission** on your institution's home page.

Self-Service features

This section describes the following features available throughout Self-Service:

- "Keyword search" on page 27
- "Help for Banner Self-Service products" on page 30
- "Web Tailor Help" on page 33
- "Site Map" on page 35
- "Display of personal information" on page 36

Keyword search

This feature places a **Search** field at the top of the Web pages. Users can enter one or more words and search for the pages on which those words appear. The field can contain up to 35 characters.

> **Note:** In the Cascade theme, the keyword search has been renamed to `Find a page`.

The search feature will only list pages that the user has permission to see.

Banner Web Tailor will process the search and begin returning results as the user is typing. This increases the speed in which results are returned and the accuracy of the results because the user can see the results as the search is being entered. All search results will display on the current page to ensure that the user can easily refine the search if necessary.

> **Note:** This is applicable only for Cascade.

> **Note:** You can specify the minimum number of characters that must be entered before Banner Web Tailor will process the search.

The application pages that match the search are displayed as links, sorted by the following priorities:

1. Info Text records with the label `KEYWORD` in the Web Tailor Repeating Information Text Table (TWGRINFO)
2. Menu names in the Web Tailor Repeating Menu Item Table (TWGRMENU)
3. Procedure and menu items in the Web Tailor Main Web Header Table (TWGBWMNU)
4. Info Text that does not have the label `KEYWORD` in the Web Tailor Repeating Information Text Table (TWGRINFO)

Note: The **Search** field does not appear on the home page. Users must logon before Web Tailor can determine which pages they have the authority to access.

The results page will not show duplicate entries or links to any web pages not used by your institution.

You can include custom Web pages or additions to pages by adding them to your local Info Text with the label KEYWORD. This allows you to customize the search so that specific pages are returned for specific search strings since Info Text matches are the highest priority. For example, if you want the search string Location to also return an address update page, you can enter Location for the address update page as an Info Text Keyword entry.

The number of results returned can be customized in the Web Tailor parameter MAXSEARCHRESULTS. You can change this value on the Web Tailor Parameters page.

Note: Currently, the search will only recognize the first word search string as a single item. Multiple words will be treated as one string, so a search for Home Address will search for the single phrase Home Address, not the word Home and the word Address.

Configure search functionality (Cascade theme)

To configure the minimum number of characters that must be entered before Banner Web Tailor will process the search, perform the following steps:

Note: Search functionality can be configured only when using the Cascade theme.

1. Open the serviceProperties.js file.

2. Search for the SearchConstants section of the file.

3. Update the searchStartLengthto value from the default value of 1 to the desired value.

```
var SearchConstants = {
                searchStartLength: '1'
}
```
4. Save the serviceProperties.js file.

Searching by keyword

To search for a Web page using a keyword:

1. In the Default theme, select the **Search** field at the top of the page or in the Cascade theme, select the **Find a page** field.

> **Note:** The **Search** field does not appear on the home page.

2. Enter a string that pertains to the topic or page you want to see.

3. Select the **Go** button or press the Enter key.

4. If the search does not find any results, you can enter a different word or string into the **Search** field and try again.

   If the search does find results, a descriptive link to the page is displayed. The results are listed in priority order.

Adding keywords

To add keywords:

1. Logon to the Secure Area in Banner Self-Service.

2. Select **Web Tailor Administration** from the top links or the menu.

3. Select **Information Text**.

4. Select the package name and corresponding procedure from the pull-down list.

5. Select **Add a New Information Text Entry**.

6. At **Select Label**, skip over the pull-down list and enter `KEYWORD` in the field. You must enter it in capital letters.

7. Enter the keywords for that page in **Information Text**, separated by commas.

8. Select **Submit Changes**.

You can customize the number of results returned:

1. Logon to the Secure Area in Banner Self-Service.

2. Select **Web Tailor Administration** from the top links.

3. Select **Web Tailor Parameters**.

4. Locate the parameter `MAXSEARCHRESULTS`.

5. Set the number to the value of your choice.

6. Save your changes.

Search implementation

The package TWBKSRCH is invoked when a user submits a search request. It presents links to each page on which the word is found and a description of that page. The links are displayed in the order in which they are found.

This package searches through the Web Tailor Repeating Menu Item Table (TWGRMENU), looking for the values entered in the **Search** field.

The search is conducted in the following order:

1. Items listed on the Web Tailor Repeating Information Text Table (TWGRINFO) with the label `KEYWORD`, where TWGRINFO contains an occurrence of `%Search String%`.

2. Items listed in the Web Tailor Repeating Menu Item table (TWGRMENU), searching in the following columns:

    2.1. `TWGRMENU_NAME`

    2.2. `TWGRMENU_URL_DESC`

3. Items listed in the procedure/menu items table, the Web Tailor Main Web Header Table (TWGBWMNU), searching in the columns:

    3.1. `TWGBWMNU_DESC`

    3.2. `TWGBWMNU_PAGE_TITLE`

    3.3. `TWGBWMNU_HEADER`

4. Items in the Web Tailor Repeating Information Text Table (TWGRINFO), searching on items that are not marked with the label `KEYWORD`.

Help for Banner Self-Service products

A system of help messages is now stored as Info Text with the label `HELP`. It was originally stored in the Self-Service products in static HTML files. By changing it into Info Text, it will be much easier for non-technical users to customize and maintain.

As with all other rows, Info Text Help records are stored in Web Tailor as Baseline. Info Text rows created by your institution using Web Tailor are stored as Local. You cannot modify Baseline rows. Info Text that is stored as Local is displayed first. If there are no rows stored as Local, the Baseline rows are displayed. This ensures that your institution's custom help will not be overwritten with each Banner upgrade.

This help is displayed when the user selects the Help link in the upper right-hand corner of the Web page. The help text appears in a new window, and includes an Exit link at the bottom of each window.

A CSS for global-level help, `web_defaulthelp.css`, is provided. It will be used in addition to the CSS defined for that page.

This style sheet will be present at all times in addition to the CSS defined for the application pages. It defines any additional styles necessary to present help text to the user.

Creating and modifying help

To create or modify help text:

1. Logon to Banner Self-Service, entering the Secure Area.

2. Access **Web Tailor Administration**.

3. Select **Information Text**.

4. Choose the web page where you want the text to appear from the pull-down list and select **Customize Information Text**.

5. If you want to modify existing Info Text, select its label. To create new Info Text, select **Add a New Information Text Entry.**

6. If you are modifying existing Info Text, it will be displayed. Otherwise, select `HELP` from the pull-down list.

7. Enter the text in the **Information Text** field or modify the text being displayed.

> **Note:** You cannot include an image with Help Info Text.

8. If you have any information to add that you do not want to appear in the help, add it in the **Comments** field.

9. Save your changes.

10. To enter an additional paragraph of text, select **Add a New Information Text Entry**, and continue from Step 6.

Separating help text into paragraphs

Web Tailor treats Info Text with the label `HELP` differently than other kinds of Info Text.

For Info Text with other labels, Web Tailor concatenates all the fields of text, displaying it as one paragraph. To separate it into multiple paragraphs, you must use a <p> tag to indicate the end of each paragraph.

Web Tailor puts Info Text labeled `HELP` into separate paragraphs if you enter it in separate fields with different sequence numbers. You do not need to use an HTML tag to do this.

Setting up a CSS for help

> **Note:** You must have the Web Tailor Administrator role to set up a CSS.

To use a CSS to format a particular help Web page:

1. Logon to the Secure Area.

2. Select **Web Tailor Administration**.

3. Select **Web Menus and Procedures**.

4. Choose the name of the procedure that generates the Web page for which you want to customize help from the pull-down list, and select **Customize Web Menu or Procedure**.

5. Enter the location of your help files in the **Help URL** field (this populates `twtvmodu_help_url`).

6. Select **Submit Changes**.

To use a CSS to format all the help pages of a particular module:

---

1. Logon to the Secure Area.

2. Select **Web Tailor Administration**.

3. Select **Web Modules.**

4. Choose a module from the pull-down list and select **Customize Module**.

5. If you want to replace Info Text help with a static HTML page, enter the location of your file in the **Help URL** field (populates `twtvmodu_help_url`).

    If you want to use Info Text help, clear this field.

6. Enter the location of the CSS that will determine the format of your help text in the **Help CSS** field (populates `twtvmodu_help_css`).

7. Select **Submit Changes.**

To use a CSS to format all the Help Info Text in all your Self-Service products:

1. Logon to the Secure Area.

2. Select **Web Tailor Administration**.

3. Select **Global User Interface Settings**.

4. If you want to replace Info Text help with a static HTML page, enter the location of your file in the **Module Help URL** field.

    If you want to use Info Text help, clear this field.

5. Enter the location of the CSS that will determine the format of your help text in the **Help CSS URL** field.

6. Select **Submit Changes**.

Help implementation

The following columns are provided for storing the locations of CSSs:

1. Package/procedure-level help (`TWGBWMNU_HELP_CSS`)

2. Module-level help (`TWTVMODU_HELP_CSS`)

3. Global- or system-level help (`TWGBGLUI_HELP_CSS`)

The procedure `twbkwbis.p_opendoc` will search the columns in the order listed.

In addition, the same three levels of customization exist for clients who choose to use their own static HTML help. Prior releases did not have this level of control for static HTML.

TWBKWBIS.P_OPENDOC

This procedure has a parameter, `suppress_header`, which identifies the page as being a help page. This allows the page to be formatted according to the Help CSS instead of the same CSS that formats the rest of the page. It looks for the help URLs in order:

First, `twbkwbis.p_opendoc` looks to see if the module/procedure-level help URL (`twbklibs.twgbwmnu_rec.twgbwmnu_help_url`) is not null. If that is the case, it will display the help as defined by that URL.

If the module/procedure-level help URL is not populated, `twbkwbis.p_opendoc` looks to see if there are any Help Info Text records available. If so, then they will be displayed in a help window.

If the Help Info Text records do not exist, `twbkwbis.p_opendoc` now looks to see if the module-level help URL (`twbklibs.twtvmodu_rec.twtvmodu_help_url`) is populated. If that is the case, it will create the help link that corresponds to that URL. This is added functionality at the module level to correspond to the other three level feature enhancements to Web Tailor.

If the module help link does not exist, the system will check to see if the global help link (`twgbglui_rec.twgbglui_help_url`) is populated. If it is, it will create the help link using that URL.

System-level CSS

The CSS for Help text is `web_defaulthelp.css`. The procedure `twbkbwis.p_opendoc` searches for this CSS if it cannot find a module-level or page-level Help CSS. It applies to all the Help Info Text in all your Self-Service.

Web Tailor Help

The static HTML files that describe Web Tailor itself were not changed into Info Text.

If you modify HTML files, or if they are redelivered in a future release, the modified files must be copied to your web server. See Chapter 2, "Configuring Self-Service Banner," in the *Banner Middle Tier Implementation Guide*, for instructions on setting up HTML files on the web server.

The following are the only remaining static HTML files that contain Help text.

File Name	Description
`twbhcont.htm`	Web Tailor Contents
`twbhmenu.htm`	Web Tailor Help Menu
`twbhpmcu.htm`	Customize a Web Tailor Parameter
`twbhtero.htm`	Reorder or Customize Information Text
`twbhgrcu.htm`	Customize a Graphic Element
`twbhitro.htm`	Reorder or Customize Menu Items
`twbhmocu.htm`	Customize a Module

File Name	Description
twbhpmfm.htm	Create a New Web Tailor Parameter/Customize the Selected Web Tailor Parameter
twbhuifm.htm	Customize Global User Interface
twbhgrfm.htm	Create a New Graphic Element /Customize the Selected Graphic Element
twbhjclr.htm	Obsolete
twbhmofm.htm	Create a New Module/Customize the Selected Module
twbhrufm.htm	Customize Web Rules
twbhwmcu.htm	Customize a Web Menu or Procedure
twbhindx.htm	Web Tailor Online Help Index
twbhlocu.htm	Customize Login Return Location
twbhovcu.htm	Customize a Web Tailor Override
twbhtecu.htm	Select Information Text to Customize
twbhwmfm.htm	Create a Web Menu or Customize an existing one
twbhitcu.htm	Select Menu Items to Customize
twbhlofm.htm	Create a New Return Location/Customize the Selected Return Location
twbhovfm.htm	Create a New Web Tailor Override/Customize the Selected Web Tailor Override
twbhtefm.htm	Create a New Information Text Entry/Customize the Selected Information Text Entry

Each Web Tailor menu item is documented online. You access Help by selecting **Help** from any Web Tailor page.

You may also customize Web Tailor's help files. As the *Upgrade Guide* for Web Tailor instructs, create a directory on the web server for the Web Tailor help pages (wtlhelp). Then customize the files using either of these methods:

• Make changes to the pages in the existing directory by customizing the HTML. Using this method, the context-sensitive links are preserved. However, when the Help is upgraded, the customized files are overwritten.

• Create a directory at the same level as wtlhelp for your custom help pages -- give the directory a unique name, such as myhelp. Copy the pages you wish to modify from wtlhelp into the new directory, and make your changes. When the next release of Web Tailor calls for Help to be installed in your wtlhelp directory, your customized pages in your myhelp directory will be preserved. If you want your new help pages to be context-sensitive, go to the corresponding application page in Web Tailor's Web

Menus and Procedures, and change the Help URL to point to your `myhelp` directory instead of `wtlhelp`.

⚠️ *Warning! If you choose the second method, the links to the Contents and Index for Web Tailor at the top of each page will not open your customized pages, because they point to* `wtlhelp`*. If you require the Contents and Index functions, you must open the HTML pages for the Contents and Index and change the link to point to your new directory.*

You may also disable the help if you choose. To do this, delete the Help URL in Web Menus and Procedures for the appropriate page.

The delivered help is stored as Baseline. Any help you create is stored as Local, so your changes will not be overwritten by later releases.

Site Map

Users go to a Site Map when they are looking for a specific piece of information from crowded menus. The Site Map is a simple list of links for all the web pages in the Self-Service product, which can help the users get a feel for the information architecture and how to find what they want.

The Site Map appears on a separate web page accessible by selecting the Site Map link. It contains a list of all the items on all the menus in your installation. There are no fields on it, only links to the various web menus and pages.

The items on the Site Map are indented to show how the pages reside under the menus and submenus. The more an item is indented, the lower it appears on the menu tree.

The Site Map shows up to three levels of menu items, although you can define any number of sublevels for your menus. Initially the Site Map displays two levels in a three-column display. Select the **Show Details** button to see three levels in a two-column display and select the **Hide Details** button to return to a three-column display.

Site map links that contain submenu items will take you to a menu page. Items that are application pages will take you directly to that application page.

All users will have access to the Site Map, but they will see only links to pages they have permission to access.

📝 **Note:** Bottom-of-the-page links do not appear on the Site Map.

Using the Site Map

To see the Site Map, click the **Site Map** link.

📝 **Note:** The Site Map cannot be accessed from the home page because access to the links on the Site Map can only be determined after the user logs on.

To change the look and feel of the Site Map, update the CSS that controls its format, `web_defaultapp.css`. The following CSS classes control the presentation of the Site Map items:

- `-sitemaplevel1` controls first level (Main Menu) items

- `-sitemaplevel2` controls second level items

- `-sitemaplevel3` controls third level items.

To add items to the Site Map, simply add them to the appropriate menu item in Web Tailor. They will appear automatically on the Site Map. For more information about adding items to a menu, refer to the *Banner Middle Tier Implementation Guide*.

Site Map implementation

The TWBKSITE package contains the procedure that generates the Site Map, `twbksite.p_dispsitemap`. It searches through the Web Tailor menu tree and display all the links to the pages on one page. This package contains the following public items:

- A procedure, `twbksite.p_dispsitemap`, which displays the Site Map

- A function, `getsitemap_c`, which populates a PL/SQL table with an entire menu when passed the name of that menu

Display of personal information

You can display specific information for the current end-user at the top of the Web pages that user is accessing.

For example, you can display:

- The student's name

- The current term

- The date

- The time

You can have a maximum of four lines of information in the headers.

The information that appears in the headers depends on the Self-Service product the person is using. For example, the information the person sees in Advancement Self-Service would be different from what they see when they access Employee Self-Service.

Web Tailor provides the mechanism for displaying the information; the individual products determine what information is shown to the users. For instructions on displaying specific information, please refer to the product-specific documentation.

The style `<DIV class="staticheaders">` supports this feature. You can add various attributes to modify the look and feel of the text.

For example, adding the attribute `text-align` with the value `right` will right-justify the text in the headers:

```
DIV.staticheaders {
text-align:right;
}
```

To prevent this information from appearing on your institution's web pages, you can add the `attribute` display with the value `none`, for example:

```
DIV.staticheaders {
display:none;
}
```

Printing is controlled by a different CSS, `web_defaultprint.css`. You can keep the static header information from appearing on printed Web pages by including the following in `web_defaultprint.css`:

```
DIV.staticheaders {
display:none;
}
```

Institution identifier

If your institution is using Banner's Multi-Institution Functionality (MIF), an institution identifier can be displayed in the header of all Self-Service web pages. This identifier will display only if your Banner implementation has been set up for MIF, and then only if the new Web Tailor HEADERDISP parameter is set to `Y`. (The HEADERDISP parameter can be set on the Customize a Web Tailor Parameter page.)

When the institution identifier appears, it indicates the institution code, as in this example: *Your current institution is OCC.*

If the web page displays user ID and date in the header, the institution identifier appears directly below the date.

Printing web pages

Creating a printer-friendly page

This CSS is present at all times, in addition to the CSS defined for each page. It provides support for printing from the Web by helping the other CSS handle page breaks, page boxes, and media dependencies. It suppresses header links, bottom links and images present in header area, so the only thing that prints is the content of most interest to the user.

If you have to make modifications to the `web_defaultprint.css`, keep in mind that the CSS will only be used when users invoke it with the Print option from the File menu of

the browser. The browser picks up the `web_defaultprint.css`, the Self-Service products do not.

Printing without Info Text

Web Tailor can be set up to with a default to print a page without its Info Text. This may be useful in situations where there is a lot of data and you want to reduce the number of pages you will be printing.

To suppress printing Info Text, edit `web_defaultprint.css`, adding the following:

```
DIV.infotextdiv {
display: none
}
```

Using Web Tailor

Web Tailor lets you build, customize, and modify the pages in your institution's Web systems. It is a component of every Banner™ Self-Service application.

Not every Self-Service application uses all of the pages of Web Tailor -- for example, the Advancement Self-Service Rules page is used only with Advancement Self-Service.

Although Web Tailor is a separate product, its effects appear only when used in conjunction with Banner Self-Service. All of the applications require both Web General and Web Tailor.

Keep in mind that there are three levels of settings maintained in Web Tailor:

1. *Global* - applies to all the Self-Service products

2. *Module* - applies to a single module, e.g., Student Self-Service

3. *Procedure* - applies to a single procedure, e.g., `bwgkomar.P_SelectMtypUpdate` (Update Marital Status)

This chapter describes the following Web Tailor features:

- "Web Tailor menu" on page 40

- "Web menus and procedures" on page 45

- "Menu items" on page 49

- "Info Text" on page 52

- "User roles" on page 55

- "Web rules" on page 56

- "Web modules" on page 62

- "Global user interface settings" on page 63

- "Web Tailor parameters" on page 65

- "Graphic elements" on page 74

- "Login return location" on page 75

- "Web Tailor overrides" on page 76

- "LDAP administration" on page 77

Web Tailor menu

To get to the Web Tailor menu, select **Secure Area** and logon using your ID and password. The Main Menu appears. Select **Web Tailor Administrator** from the Navigation Bar links or from the menu, and the Web Tailor Menu appears.

It includes the following options:

* Web Menus and Procedures

* Menu Items

* Information Text

* User Roles

* Web Rules

* Web Modules

* Global User Interface Settings

* Web Tailor Parameters

* Graphic Elements

* Login Return Location

* Web Tailor Overrides

* Advancement Self-Service Rules (Advancement Self-Service)

* Finance Site Preferences (Finance Self-Service)

* LDAP Administration

* Passed Planned Moves Exist (Advancement Self-Service)

* Channel Maintenance (only if you are using Banner Channels for Luminis®)

* Channel Preference (only if you are using Banner Channels for Luminis)

Web Tailor Utility Report pages

These six reports let you review Web menus/procedures, menu items, information text, and role settings being delivered for specific Web products through a web browser instead of SQL*Plus.

The *compare* reports display the production baseline, local, and/or install data on the same web page. This allows you to decide which of the updates you want to apply to your production system, or which ones you might want to update manually, depending on the customizations you have done. Use these reports when you install the Self-Service upgrades.

The *view* reports allow you to view the contents of four specific tables via the web page. Use these reports anytime, or when you install the Self-Service upgrades.

Compare reports

Three Web-based reports delivered with Web Tailor display the contents of the twgbwmnu, twgrinfo, twgrmenu, and twgrwmrl tables, enabling administrators to compare new delivered data with their existing data:

- `cmprmenu.sql` (twgrmenu table)

- `cmprinfo.sql` (twgrinfo table)

- `cmprwmnu.sql` (twgbwmnu and twgrwmrl tables)

The scripts are located in the Web Tailor Plus directory. Logon to SQL*Plus as the `wtailor` schema owner and run the scripts to create the packages. These scripts will create the public synonyms and grants the user needs to run them on the Web. To get to the reports on the Web, the user must point his Web browser to the following:

    http://<servername>/<dad>/<listener>/cmprwmnu.p_start

where:

`<servername>` is the name of your Web server running Oracle 10g

`<dad>` is the Data Access Descriptor; use `pls` for iAS

`<listener>` is the name of the Apache HTTP listener defined by your systems administrator

You can substitute `cmprinfo` or `cmprmenu` in place of `cmprwmnu`.

The compare reports are accessible only by users defined as Web Tailor Administrators. These reports are part of a menu, which is delivered as *Not Enabled* since the reports are not a standard option. To enable the menu, perform the following steps:

1. Login as a user that has a Web Tailor Administrator role.

2. Click the Web Tailor Administration tab.

3. Select Web Menus and Procedures

4. Enable bmenu.P_WTUtilityReportsMnu.

   Note: If the reports have been compiled, they are accessible by a Web Tailor administrator by entering the URL directly.

Compare reports

Procedures are as follows:

1. Select the name of the product

2. Select the type of table you wish to query.

The *wtailor* user *must* have select privileges on the specified tables. Please ensure that any necessary synonyms or grants are in place before selecting from a non-`wtailor` table. If you choose to query other tables, please be sure to enter the table name correctly and select the appropriate table owner.

3. Select the Owner.

View reports

Three other Web-based reports delivered with Web Tailor display the contents of the twgbwmnu, twgrinfo, and twgrmenu tables, enabling administrators to view production, install, or other local tables. They are similar to the comparison reports, but only look at one table at a time. These scripts are:

*   `viewmenu.sql` (twgrmenu table)

*   `viewinfo.sql` (twgrinfo table)

*   `viewwmnu.sql` (twgbwmnu and twgrwmrl tables)

Logon to SQL*Plus as the `wtailor` schema owner and run the scripts to create the packages.

The scripts are located in the Web Tailor Plus directory. These scripts will create the public synonyms and grants the user needs to run them on the Web. To get to the reports on the Web, the user must point his Web browser to the following:

```
http://<servername>/<dad>/<listener>/viewwmnu.p_start
```

where:

`<servername>` is the name of your Web server running Oracle 10g

`<dad>` is the Data Access Descriptor; use `pls` for iAS

`<listener>` is the name of the Apache HTTP listener defined by your systems administrator

You can substitute `viewmenu` or `viewinfo` for `viewwmnu`.

View reports

Procedures are as follows:

1. Select the name of the product

2. Select the type of table you wish to query.

    The *wtailor* user *must* have select privileges on the specified tables. Please ensure that any necessary synonyms or grants are in place before selecting from a non-`wtailor` table. If you choose to query other tables, please be sure to enter the table name correctly and select the appropriate table owner.

    You must enter values for both copies of the parameters.

3. Select the Owner.

The rest of this section describes, in detail, the Web Tailor options.

Please refer to the product-specific user guides for the other options. For more information about channels, please refer to the Banner Channels for Luminis documentation.

Web Tailor Utility Report pages

These six reports let you review Web menus/procedures, menu items, information text, and role settings being delivered for specific Web products through a web browser instead of SQL*Plus.

The *compare* reports display the production baseline, local, and/or install data on the same web page. This allows you to decide which of the updates you want to apply to your production system, or which ones you might want to update manually, depending on the customizations you have done. Use these reports when you install the Self-Service upgrades.

The *view* reports allow you to view the contents of four specific tables via the web page. Use these reports anytime, or when you install the Self-Service upgrades.

Compare reports

Three Web-based reports delivered with Web Tailor display the contents of the twgbwmnu, twgrinfo, twgrmenu, and twgrwmrl tables, enabling administrators to compare new delivered data with their existing data:

- `cmprmenu.sql` (twgrmenu table)

- `cmprinfo.sql` (twgrinfo table)

- `cmprwmnu.sql` (twgbwmnu and twgrwmrl tables)

The scripts are located in the Web Tailor Plus directory. Logon to SQL*Plus as the `wtailor` schema owner and run the scripts to create the packages. These scripts will create the public synonyms and grants the user needs to run them on the Web. To get to the reports on the Web, the user must point his Web browser to the following:

```
http://<servername>/<dad>/<listener>/cmprwmnu.p_start
```

where:

`<servername>` is the name of your Web server running Oracle 10g

`<dad>` is the Data Access Descriptor; use `pls` for iAS

`<listener>` is the name of the Apache HTTP listener defined by your systems administrator

You can substitute `cmprinfo` or `cmprmenu` in place of `cmprwmnu`.

The compare reports are accessible only by users defined as Web Tailor Administrators. These reports are part of a menu, which is delivered as *Not Enabled* since the reports are not a standard option. To enable the menu, perform the following steps:

1. Login as a user that has a Web Tailor Administrator role.

2. Click the Web Tailor Administration tab.

3. Select Web Menus and Procedures

4. Enable bmenu.P_WTUtilityReportsMnu.

> **Note:** If the reports have been compiled, they are accessible by a Web Tailor administrator by entering the URL directly.

Compare reports

Procedures are as follows:

1. Select the name of the product

2. Select the type of table you wish to query.

   The *wtailor* user *must* have select privileges on the specified tables. Please ensure that any necessary synonyms or grants are in place before selecting from a non-`wtailor` table. If you choose to query other tables, please be sure to enter the table name correctly and select the appropriate table owner.

3. Select the Owner.

View reports

Three other Web-based reports delivered with Web Tailor display the contents of the twgbwmnu, twgrinfo, and twgrmenu tables, enabling administrators to view production, install, or other local tables. They are similar to the comparison reports, but only look at one table at a time. These scripts are:

- `viewmenu.sql` (twgrmenu table)

- `viewinfo.sql` (twgrinfo table)

- `viewwmnu.sql` (twgbwmnu and twgrwmrl tables)

Logon to SQL*Plus as the `wtailor` schema owner and run the scripts to create the packages.

The scripts are located in the Web Tailor Plus directory. These scripts will create the public synonyms and grants the user needs to run them on the Web. To get to the reports on the Web, the user must point his Web browser to the following:

```
http://<servername>/<dad>/<listener>/viewwmnu.p_start
```

where:

`<servername>` is the name of your Web server running Oracle 10g

<dad> is the Data Access Descriptor; use `pls` for iAS

<listener> is the name of the Apache HTTP listener defined by your systems administrator

You can substitute `viewmenu` or `viewinfo` for `viewwmnu`.

View reports

Procedures are as follows:

1. Select the name of the product

2. Select the type of table you wish to query.

   The *wtailor* user *must* have select privileges on the specified tables. Please ensure that any necessary synonyms or grants are in place before selecting from a non-`wtailor` table. If you choose to query other tables, please be sure to enter the table name correctly and select the appropriate table owner.

   You must enter values for both copies of the parameters.

3. Select the Owner.

Web menus and procedures

The Web Menus and Procedures option lets you create a new menu or procedure, or choose an existing one from the list by clicking its **Procedure Name**. Either choice brings you to the Web Menus and Procedures page.

> **Note:** The list order is controlled by **Display Descriptive Names** on the Web Rules page. If checked, then the list is ordered by **Description**; if unchecked, it is ordered by procedure name.

You can optionally do a text search for a web menu or procedure, and then select from the list of search results. You can search by the **Name** of the web menu or procedure, or by its **Description**.

To search for and select a web menu or procedure:

1. Type the full or partial name in the **Search by Name** field, or the full or partial description in the **Search by Description** field. Search text is not case-sensitive.

   > **Note:** If you enter search terms in both fields, **Search by Name** will take priority and the terms in **Search by Description** will be ignored.

2. You can use % as a wildcard character in either field, or _ as a single-character wildcard.

3. Select **Search**.

4. Select the web menu or procedure from the list of search results, or search again.

Web Page Fields

Item	Description
Page Name	The name of the web page you are creating or modifying. Required.
Description	The description of the page. Required.
Module	The name of the Self-Service product to which the page belongs. You can use this field to make queries, but you should not change the delivered values. If your institution creates its own entries for this table, you may select your own modules. Required.
Comments	Enter any comments about the page.
Enabled Indicator	If the check box is selected, the page can be accessed from menus and other pages. Otherwise, it cannot.
Non Secured Access Allowed	Select the check box if users can access this page without entering a user ID and password.

Web Page Caching Override	Sets the web page caching method for browsers that support it. Valid values are:
	
	`Use the system setting`
	`Allow Caching`
	`Do Not Allow Caching`

Page Title	The page title that displays at the top of the page.
Header Text	The page header.
Header Graphic	The name of the graphic you want at the top of the page.
Page CSS URL	The name of the cascading style sheet you want to use for the web page.
Map Title	The map name to use for the TITLE attribute in the `htp.mapopen` function called at the beginning of `twbkfrmt.p_printmenu`.
Help Link URL	The URL that contains the help file for this page.
Help CSS URL	The cascading style sheet you would like to use for the Help.
Print CSS URL	The cascading style sheet you would like to use for the printing of the web page.
Back Link URL:	The URL of the previous page.
Back Link Text:	The text that appears when the user places their mouse on the Back link.

Item	Description
Back Link Image:	The image you want users to select to move to the previous page.
Back Link Menu Indicator:	Select the check box if you want the back link URL to display.
Admin Secured	Indicates if the Web page is secured under the VBS and PII rules set up in Banner. Valid values are: `Y` - an Oracle username and password will be required to access the page, and the VBS and PII rules (as defined in Banner for the user's role, e.g., `Faculty`) will apply to the user's activities for the duration of the user's session. `N` - there are no special restrictions on this page.

The available roles are listed at the bottom of the page. You must select the appropriate check boxes to give different types of users access to your menu. There may be other requirements, though. Please see the specific Self-Service product documentation for information about each one.

When you customize a web menu or procedure, a local copy of the menu or procedure is created and you are actually customizing the local copy, not the baseline copy. The baseline data is shown alongside the local data, with a flag icon indicating where the data differs.

Generating menu pages

Menu pages can be easily generated by calling the existing procedure `twbkwbis.P_GenMenu`; when you use the submenu indicator on the Customize the Selected Menu Item page, that procedure is automatically used.

It is highly recommended that functional pages call `twbkwbis.P_OpenDoc` to generate the HTML document open, title, body, and header tags that begin a page. `P_OpenDoc` uses the information provided in twgbwmnu when it generates this code. The procedure `twbkwbis.P_CloseDoc` prints any bottom-of-page links associated with the page. `P_OpenDoc` also provides accessibility functionality.

Parameter `print_module_bottom-links` was added to suppress module bottom links for printing. The default value for this parameter is `true`.

All procedures which use `P_OpenDoc` to open the page *must* use `P_CloseDoc` to close the page, and vice versa, due to Web Tailor's use of HTML tables for page formatting purposes.

> **Note:** `P_CloseDoc` cannot be used in a page unless `P_OpenDoc` is also called. Please refer to any of the General Web packages for a

specific example of the `P_OpenDoc` and `P_CloseDoc` procedures (for example, `twbkwbil.sql`).

Menu items

The Menu Items option lets you create or change the items on your menus. Click Create or click the **Procedure Name** of a menu from the list.

> **Note:** The list order is controlled by **Display Descriptive Names** on the Web Rules page. If checked, then the list is ordered by **Description**; if unchecked, it is ordered by procedure name.

You can optionally do a text search for a web menu, and then select from the list of search results. You can search by the **Name** of the web menu or procedure, or by its **Description**.

To search for and select a web menu:

1. Type the full or partial name in the **Search by Name** field, or the full or partial description in the **Search by Description** field. Search text is not case-sensitive.

   > **Note:** If you enter search terms in both fields, **Search by Name** will take priority and the terms in **Search by Description** will be ignored.

2. You can use `%` as a wildcard character in either field, or `_` as a single-character wildcard.

3. Select **Search**.

4. Select the web menu from the list of search results, or search again.

Reorder or customize menu items

Baseline menu items cannot be modified. You must copy them to Local and change the Local version. For more information on Baseline and Local, please refer to "Technical Reference" on page 86

Web Page Fields

Item	Description
Sequence Number	The number that orders the display of menu or procedure items. Required.
Source	Indicates if the item is Baseline or Local. Display-only.
Link Text	The text that will appear on the menu for this link. Optional.
URL	The URL of the page that is displayed when the user selects the menu item. Required.
Enabled	Indicates whether the page is enabled or not. Y—Enabled N —Not enabled

To change the order in which the menu items will appear, change the sequence numbers and select **Reorder these Elements**.

To customize a menu item, select its link.

Customize the Selected Menu Item

A menu item can be customized by modifying the following values:

Web Page Fields

Item	Description
Associated Web Page	The package.procedure of the menu that the item appears on.
Sequence Number	The number that orders the display of menu or procedure items. Required.
URL	The URL of the page that is displayed when the user selects the menu item. The package.procedure or menu name that will be called when the user selects the item. A string of text or an HTML image tag can be specified as the hyperlink. Required.
Link Text	The text that will appear on the menu for this link. Optional.
Link Description	The description that appears below the link on the menu.
Status Bar Text	The text that appears on the Status Bar when you rest your cursor over the menu item.
Bullet Override Image	The Web Tailor name for the image at the left side of the link.
Menu Link Image	This field contains the optional image name to be displayed before the hyperlink defined in the **Link** field for menu definitions only.
Enabled	Indicates if is field is used by web procedures to determine which menu or procedure items will be displayed on the web page. For example, if you do not want users to be able to change their Emergency Contacts from the Personal Information Menu, set that item's Enabled Indicator to `N`, and the link will not appear in that menu. Valid values are: `Selected` - enabled `Cleared` - disabled Null - disabled
Database Procedure	Specifies if the menu item is a database procedure. Valid values are: `Selected` - The item is a stored procedure `Cleared` - It is a menu item, a regular URL Null - It is a menu item
Submenu Indicator	Indicates if the URL will call a Web Tailor menu. Valid values are: `Selected` - Submenu `Cleared` - Not a submenu Null - Not a submenu

Adding new menu items

There are two types of menu items, those displayed on a menu page and those displayed at the bottom of a functional page, also called `bottom-of-page links`.

It is recommended that you use the existing procedure `twbkwbis.P_GenMenu` to display menu pages. In fact, if you set a menu item's Sub(menu) indicator to checked, `P_GenMenu` will automatically be called when that link is selected. Bottom-of-page links are best displayed by calling `twbkwbis.P_CloseDoc`.

Modifying existing menu items

Whenever you add, delete, or make a menu item inactive, you should examine the menu items on the page which is being linked. Generally, if you can get from page A to page B through a bottom-of-page link, you should also be able to get from B back to A through another link. So, if you add a link from the View E-mail Addresses page to the View Addresses and Phones page, you should probably also add a link to View Addresses and Phones which will take the user to View E-mail Addresses.

Also, remember that menu items can only be accessed if the user has appropriate roles. For example, you should not move the **Display Grades** item to an Employee menu, because students would not be able to access it.

Menu items may be temporarily disabled without deleting them. This may be very useful for pages which allow student registration or employee open enrollment. When these functions are not allowed by your institution, simply clear the **Enabled** indicators. Your menu item information will be preserved for the next time the function should be available. Whenever you enable or disable menu items, make sure you find all the occurrences of the link. For example, View Addresses and Phones is available from the Personal Information menu, the Update Addresses and Phones page, and Update Addresses and Phones page.

Info Text

Info Text is a brief set of instructions or information that the user may find helpful when looking at a page. In certain cases, such as for the login procedures, Info Text may be error messages that your institution can customize. You may also specify a graphic which will call attention to the text, and HTML tags can be embedded directly into the text. The delivered data is displayed after a text graphic with the letter `i` in the center (`web_info.gif`). The text usually has a paragraph break (<p>) at the end.

Info Text is broken into multiple records, each with a sequence number and label. All records belonging to the same procedure name and the same label will be printed together in the order indicated by the sequence numbers. Most pages display Info Text only at the top of the page, and those records are usually labeled DEFAULT. However, if a page has special uses for Info Text, as the `twbkwbis.P_ValLogin` procedure does, the labels will contain text that indicates their use.

All of the labels have been used in the source code, and they should not be modified in any case. You may however, add, delete, or modify records for existing procedure name/label combinations.

Select Info Text to customize

To create or modify Info Text, select **Information Text** from the Web Tailor menu. Click Create or choose an item from the list by clicking its **Procedure Name**.

> **Note:** The list order is controlled by **Display Descriptive Names** on the Web Rules page. If checked, then the list is ordered by **Description**; if unchecked, it is ordered by procedure name.

You can optionally do a text search for a web menu, and then select from the list of search results. You can search by the **Name** of the web menu or procedure, or by its **Description**.

To search for and select a web menu:

1. Type the full or partial name in the **Search by Name** field, or the full or partial description in the **Search by Description** field. Search text is not case-sensitive.

   > **Note:** If you enter search terms in both fields, **Search by Name** will take priority and the terms in **Search by Description** will be ignored.

2. You can use `%` as a wildcard character in either field, or `_` as a single-character wildcard.

3. Select **Search**.

4. Select the web menu from the list of search results, or search again.

Reorder or customize info text

With the exception of `DEFAULT` all labels are triggered by the logic programmed into the page.

The program that generates the Web page has specific locations where it may look to display Info Text with a specific label. You can customize the text, but you cannot change the place on the page that the text is displayed.

To modify this text, Select **Copy Baseline entries to Local**, then select the label of the text you want to change. The Customize the selected Information Text Entry page appears.

Make your changes and save them.

Info Text has been written to be used with all the Self-Service products. If your institution has not licensed all the products, you may want to customize some of the messages to refer to only those which you have.

⚠️ ***Warning!*** *It is very easy to affect the entire page's appearance by making an HTML error in the Info Text. Please test your changes thoroughly.*

Web Page Fields

Item	Description
Associated Web Page	The package.procedure for the Web page.
Sequence Number	The sequence number indicates in which order this text will appear, if more than one item is to be displayed.
Label	Identifies the individual record of Info Text that will be displayed in a single area of the web page. Multiple records of information can be displayed anywhere that Info Text appears on any web page. Required. Your labels should consist of capital letters only.
Information Text	The text that should display for this record. Text may include HTML tags such as italics, line breaks or images, although it is recommended that you minimize your use of HTML for the sake of simplicity, and because it is difficult to diagnose page display problems caused by incorrect HTML in Info Text. You may continue text in the next record with the same label if you need more space. Be sure to set up the sequence numbers to display your text appropriately.
Image	The file name of the image to be displayed before the Info Text.
Comment	A comment to describe the record.

Info Text in SSB footer

Beginning with the 8.4 release, you can add information text at the bottom of a page. Previously, the user could enter information text only at the top of the page. There were occasions when additional information was required on a page, but adding this information at the top of the page cluttered it.

1. Users can now create info text for any pages desired using `FOOTER_BEFORE_LINK or FOOTER_AFTER_LINK` (the link that is the separator is the list of hyperlinks typically listed at the bottom of a web page).

2. Users can now add an info text for `GLOBAL_FOOTER` or `GLOBAL_FOOTER_xxx` associated with the `twbkswbis.P_WWWLogin` procedure that would then be displayed on all pages. Here xxx refers to the package name that starts with xxx. For example, `GLOBAL_FOOTER_BWP` would be displayed on all the payroll pages (since

those packages all begin with BWP, BWS would be for student), and then `GLOBAL_FOOTER` which would be displayed on all pages.

3. The `FOOTER_BEFORE_LINK` and `FOOTER_AFTER_LINK` would be info text created on the specific page where the `GLOBAL_FOOTER_xxx` and `GLOBAL_FOOTER` would be created on the `twbkwbis.P_WWWLogin` page.

Note: Due to the nature and flexibility of this infotext, you cannot use the image on the Customize Info Text. Instead of this, you will have to include the image in the infotext.

Coding a link to open a new browser window

You can manually add `target="_blank"` to HTML `<A HREF>` tags in link text entries as needed to make the links open in a new browser window. The example below demonstrates how this works.

This link will open the specified page in the same window:	`<A HREF="http://www.ssa.gov/`
	`replace_sscard.html">Social Security`
	`Online</A>`

This link will open the specified page in a new browser window:	`<A HREF="http://www.ssa.gov/`
	`replace_sscard.html"`
	`target="_blank">Social Security`
	`Online</A>`

User roles

User roles in Self-Service determine what menus are displayed after logging on and what each person has access to. Users can see and access only those menu items that their roles authorize them to see. A user may have more than one role.

Note: Do not confuse the concept of web user roles with the Oracle roles used in Banner security. For information about Banner security's use of roles, see the *Banner Security Administration Handbook.*

To assign roles to a person, select **User Roles** from the Web Tailor menu. Enter the person's user ID, then select **Submit**.

Select the check boxes that correspond to the roles you want that user to have, then save your changes.

For more information and a partial list of roles used in Banner Self-Service, see <u>"Web roles" on page 99</u>.

Web rules

Web rules are global settings. They affect all your Self-Service applications.

> **Note:** The programming logic in Web Tailor that authenticates user credentials is bypassed if your institution uses LDAP to authenticate Banner Self-Service. Any web rules that pertain to user logons will be ignored.

To define and customize web rules, select **Web Rules** from the Web Tailor Menu.

Web Page Fields

Item	Description
CGI-BIN Directory	Enter the path of the Web server (CGI-BIN) directory. This column specifies the Oracle Apache PL/SQL Agent service name and OWA program name that is used to construct URLs. This can also specify a third-party Web server's CGI binary directory that will execute the Oracle Apache PL/SQL Agent program. Required.
CGI-BIN Admin Directory Suffix	Contains the suffix of the administrative path for the Oracle DAD. This is used when the Oracle FGAC utility is being used in Banner Self-Service.
	VBS allows you to restrict access to what users can see and update on various Web pages. You must set up the restrictions in Banner General. For more information, please refer to the *Banner Data Security Handbook*.
Web Timeout in minutes	Enter the number of minutes without activity before the current session is disabled. Required.
	Beginning with Release 8.4, the Web Rules page enables the setting of timeout values by role. This will enable users that have certain roles to have a longer timeout period than the one system value that has been used in the past. The user timeout value will now be the greater of the system value or any of the roles that are associated with the individual user. If the system timeout is 30 minutes and a student role has an override of 15 minutes and employee override of 45 minutes, a student only role person will have a timeout of 30 minutes, and an employee who is also a student will have 45 minutes.

Item	Description
Maximum Number of Login Attempts	Enter the maximum number of incorrect logins allowed before the user's Web account is disabled. This applies to both internally-authorized Banner users (students, faculty, employees) and external users (admissions applicants, alumni). When this limit is reached, the Web ID record is locked and the person is not allowed to access the Web system. Required. When an authorized Banner user's account is locked, the **PIN Disabled** indicator on the Third Party Access Form (GOATPAC) is automatically selected. To unlock the account, clear the check box. When an external user's account is locked, the **Web Access Locked indicator** on the Change PIN block of the Electronic Application Process Form (SAAEAPS) is automatically selected. To unlock the account, clear the check box. If your institution uses LDAP to authenticate user logons, this value is ignored.
Date Display Format Mask	Enter the date format mask you would like to use. Format masks are used to convert Oracle values to character strings that can be printed. For example, *MM/DD/YYYY* prints 03/13/2004. The string must be a valid ORACLE date format. Required.
Date Input Format Mask	Choose the date input format mask you would like to use from the pull-down menu. Format masks are used to convert Oracle values to character strings that can be printed. For example, *MM/DD/YYYY* prints 03/13/2004. This string must be a valid ORACLE date format. Required.
Time Format Mask	Enter the time format mask you want to use. Format masks are used to convert Oracle values to character strings that can be printed. For example, *HH:MI:SS* prints 08:30:59. This string must be a valid ORACLE time format. Required.
Start Page	Choose, from the pull-down menu, the page that will display immediately after a successful login for the Self-Service applications running in non-integrated mode. The delivered value is `bmenu.P_MainMnu`. Required.
Start Page is a Dynamic Menu	Indicates if the Start Page is a dynamic menu. Valid values are: `Selected` - Web Tailor generates the Start Page dynamically. `Cleared` - The Start Page is a static html file.

Item	Description
PIN Expiration in days	When a PIN is created or reset, a new expiration date is calculated for it. This value indicates how many days will elapse before the PIN expires. If your institution's policies do not call for PIN expiration, leave this field blank. If your institution uses LDAP to authenticate user logons, this value is ignored.
Display Usage Page	Select this check box to display *Acceptable Usage* page when the user first logs into Banner Self-Service. Your institution may place general information on this page about the user's responsibilities for changes made to their records via the Web. Use Web Tailor to specify the contents of the page. The user can accept the Terms of Usage, or exit. If the user accepts, the system selects **Web Access Terms Accepted** on the Third Party Access Form (GOATPAC) in Banner General. If your institution wishes to change the contents of the Terms of Usage page later, you can clear the **Web Access Terms Accepted** check box on GOATPAC. Next, select the **Display Usage Page** check box in Web Tailor to display the new page to every validated user the next time they logon.
Display Descriptive Names	Select this check box to display the information entered in the **Description** field in the Web Tailor pull-down lists. Valid values are: Selected - display the information in the **Description** field Cleared - display the name of the package.procedure
Enable Web Page Caching	Indicates whether Web caching is enabled for the entire system. Valid values are: Selected - Web page caching is enabled Cleared - Web page caching is disabled

Item	Description
Use HTTP Redirection After Login	Indicates how you want to determine the page to which users are sent when they are redirected. Valid values are: `Selected` - Web Tailor uses the URL specified in the OWA redirect utility to determine the page users will be sent to when they are redirected. `Cleared` - Web Tailor uses an HTML `META` refresh tag to specify the page. Oracle recommends not using the OWA redirect utility, so the check box should be cleared.
Java Classpath	Obsolete
Document Type Definition	Contains high level information about the Document Type Definition (DTD) that the web page was coded to. The value delivered in this column is: `HTML PUBLIC`. The HTML string indicates the top level element type declared in the DTD. The PUBLIC string indicates whether the identifier is a publicly accessible object (PUBLIC) or a system resource (SYSTEM) such as a local file or URL. HTML DTDs are specified by PUBLIC identifiers.
Document Type Definition FPI	Contains the Formal Public Identifier (FPI) portion of the DTD. The value delivered in this column is: `-//W3C//DTD HTML 4.01 Transitional//EN`. The `Registration` is indicated by a plus (+) or minus (-). A plus indicates that the organization name that follows is ISO-registered; a minus indicates that it is not. The IETF and W3C are not registered ISO organizations; they use a minus. The W3C string is the `Organization`. This is the OwnerID - a unique label indicating the name of the entity or organization responsible for the creation and maintenance of the artifact (DTD, etc.) being referenced by the `!DOCTYPE`. The IETF and W3C are the two originating organizations of the official HTML DTDs. `DTD` is the Public Text Class - the type of object being referenced. There are many different keywords possible here, but in the case of an HTML DTD, it is "DTD" - a Document Type Definition.

Item	Description
	`HTML 4.01 Transitional` is the Public Text Description - a unique descriptive name for the public text being referenced.
	`EN` is the Public Text Language; the natural language encoding system used in the creation of the referenced object. It is written as an ISO 639 language code (2 uppercase letters.) HTML DTDs are usually written in English (`EN`).
Document Type Definition URL	Contains the URL portion of the DTD.
	Together, the previous two columns and this one make up the `!DOCTYPE` tag used by Self-Service applications. If you do not alter the delivered values, the `!DOCTYPE` tag for each Web page will look like this:
	`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">`
	This tag is used almost exclusively for validation purposes.
	Validation occurs when someone (usually a Web developer) takes the static HTML that is generated for each Web page and inputs it to a validation tool. The tag forces the tools to validate according to the HTML 4.01 Transitional DTD which includes all elements and attributes of HTML 4 Strict but adds presentational attributes, deprecated elements, and link targets.
	The `!DOCTYPE` statement tells the browser how to interpret your page. Based on how your DTD is built, the browser will switch between *Standard* and *Quirk* mode.
	In *Standard* mode, the browser will render the HTML in compliance with the latest standards. In *Quirk* mode, the browser will render the HTML as though it were written for older, quirkier browsers. Quirk mode allows workarounds for older browsers to work in the latest browsers.
	Since the Self-Service Web pages need to work on both older and newer browsers, the values delivered in the three new columns will make up an `!DOCTYPE` tag that will force the browsers to render in Quirk mode.
	From Web tailor 8.3 onwards to support cascade theme on all browsers use the URL http://www.w3.org/TR/html4/loose.dtd.

Item	Description
Language ISO 639-1	The value in this field is assigned to the LANG attribute in the HTML tag on all Self-Service web pages. This attribute specifies the base language of an element's attribute values and text content. Values should conform to ISO 639 standards.

Web modules

You can customize each Self-Service product individually by setting up rules that apply only to that product.

> **Note:** If your are using the Cascade theme, modules cannot be customized on an individual basis. Cascade cannot be turned on or off on a module by module basis; it is applied to all Self-Service Banner modules.

Select **Web Modules** from the Web Tailor Menu. Choose the application you want to customize (or Common to change them all), and select **Customize Module**.

Web Page Fields

Item	Description
Module Code	The code that represents the Self-Service product.
Module Description	The corresponding description. Required.
Module CSS URL	The URL that points to the location of the module-wide Cascading Style Sheet.
Module Help URL	The URL that points to the location of the application's html Help files. Help text is in Info Text, not HTML files. Use this only if you want to override the delivered Info Text help system.
Help CSS URL	The URL that points to the location of the Cascading Style Sheet (CSS) that determines the look-and-feel of all the application's help text.
Global Menu Bottom Links	Specifies the link that appears at the bottom of each of the application's pages for the module. It is a web page, in package.procedure format. Global Menu Bottom Links do not display on menu pages in the Cascade theme. Global Menu Bottom Links will only display on menu pages that use the Default theme.
Current Release Number	Specifies the release number that will appear at the bottom of all the application's pages.
Display Exit Link	Indicates if the Exit link should appear on the application's pages.
Back Image	Specifies the image that will appear before the Back link, if any.

Global user interface settings

From the Web Tailor Menu, select **Global User Interface Settings**.

The User Interface in Banner Self-Service includes a table, TWGBGLUI, which stores settings that affect the look and feel of all dynamic Web pages.

In Global User Interface Settings, you will find the following fields:

Web Page Fields

Item	Description
System or Institution Name	Enter the name of your system and institution that you want to appear at the top of every Banner Web page. Required.
Header Image	Enter the name of the image to be displayed at the top of each page. This is not the actual file name of the image, but the name specified for it in Web Tailor. Select the link to see the actual image in a pop-up window.
Name of Main Menu	Enter the name of the main menu as you want it to appear in other systems like Luminis Platform.
CSS URL	This is the name of the Cascading Style Sheet (CSS) that controls the look-and-feel of all application pages throughout the system. This style sheet is invoked when no page-specific or module-specific style sheet has been defined. It must be a URL.
Help URL	Enter the URL of the help page to go to when a user selects the help text. Only enter this if you want to override the delivered Info Text help system and use your own static HTML help instead.
Help CSS URL	Enter the URL of the CSS that specifies the format of the help text.
Error Image	Enter the name of the Error icon image. This is not the actual file name of the image, but the name specified for it in Web Tailor. This image will be displayed with all error messages. This field is optional, but highly recommended. A default graphic is provided. Select the link to see the actual image in a pop-up window.
Warning Image	Enter the name of the Warning icon image. This is not the actual file name of the image, but the name specified for it in Web Tailor. This image will be displayed with all warning messages. This field is optional, but highly recommended. A default graphic is provided. Select the link to see the actual image in a pop-up window.
Required Image	Enter the name of the Required icon image. This is not the actual file name of the image, but the name specified for it in Web Tailor. This image will be displayed with all required fields. This field is optional, but highly recommended. A default graphic is provided. Select the link to see the actual image in a pop-up window.

Item	Description
Back Image	Enter the name of the Back icon image. This is not the actual file name of the image, but the name specified for it in Web Tailor. This image will be displayed as a link to the previous page. This field is optional, but highly recommended. Select the link to see the actual image in a pop-up window.
Submenu Image	Enter the name of the Submenu icon image. This is not the actual file name of the image, but the name specified for it in Web Tailor. This image will be displayed to the left of full-page menu items that point to a submenu. This field is optional, but highly recommended. Select the link to see the actual image in a pop-up window.
Application Page Image	Enter the name of the Application Page icon image. This is not the actual file name of the image, but the name specified for it in Web Tailor. This image will be displayed to the left of full-page menu items that point to an application page. This field is optional, but highly recommended. Select the link to see the actual image in a pop-up window.
Homepage URL	This field indicates the URL to be associated with the Powered By logo at the bottom of each web page. Select the link to see the actual image in a pop-up window.

Web Tailor parameters

Web Tailor parameters control a wide variety of Self-Service functions, such as credit card payment settings, Luminis integration settings, and the CSS to use when printing web pages.

Select **WebTailor Parameters** from the Web Tailor Menu to access the Customize a Web Tailor Parameter page.

The Web Parameters table includes several records that enable Luminis Platform integration. Examine the parameters with an asterisk (*) carefully in the `igbparm.sql` script before running `twupprod.sql`. The Luminis Platform properties mentioned here are set in the configuration file located at $`CP_ROOT/config/site-site.directory`.

Web Page Fields

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
* CPBASEURL	Fully qualified base URL for Luminis Platform system. Example: `http://machine.university.edu:8080`
* CPCOOKIEDOMAIN	The domain in which the shared global cookie is valid. The entry should contain a leading dot (`.`). The Self-Service and Luminis web servers must both be in this domain for successful maintenance of the global cookie. Luminis property name: `sctip.sharedCookieDomain` Example: `.university.edu`
* CPCOOKIENAME	Name of the shared global cookie to be used by the application and Luminis. This should match the name of the cookie defined in the Luminis configuration files. Luminis property name: `sctip.sharedCookieName` Example: `CPSESSID`
ALLOWSSNLOGIN	If `Y`, users can enter their Social Security Numbers in place of their IDs when logging in. For more information, see "Login options for user IDs" on page 83.
AUDITUSERID	This parameter controls what user ID is stored when data is changed through Self-Service. For more information, see "Capture Banner User ID" on page 84. `BANNERID` - the Self-Service user's Banner logon ID is captured and stored where possible. `WEBUSER` - the generic Self-Service user ID specified by the `WEBUSER` parameter is captured and stored.
CCMAXPIPESIZE (Obsolete)	This parameter is obsolete as of the 8.3 release. The size of the DB pipe used for Credit Card processing. Example: 8192.

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
CCREADPIPETIME (Obsolete)	This parameter is obsolete as of the 8.3 release. The amount of time, in seconds, before pipe messages time out. Example: 300
CCRESPONSEPIPETIME (Obsolete)	This parameter is obsolete as of the 8.3 release. The amount of time, in seconds, that the Self-Service applications should wait for responses from the credit card payment server. Example: 300
CCUSEADDRESS (Obsolete)	This parameter is obsolete as of the 8.3 release. Specifies if the system should use the address information for credit card verification. Valid values are: Y - Use the address information for verification N - Do not use the address information for verification
CCWRITEPIPETIME (Obsolete)	This parameter is obsolete as of the 8.3 release. The time, in seconds, required for the information from the pipe to be written to the database. Example: 5
CPCOOKIEPATH	This specifies the subset of URLs on the origin server to which the cookie applies. This normally will not require modification. Luminis property name: `sctip.sharedCookiePath` Example: /
CPINUSE	This parameter is no longer used.
CPPASSWDEXP	This specifies if you want to use the PIN expiration feature in Banner or in Luminis Platform. Valid values are: Y - User IDs and PINs will be authenticated using the settings in the Third Party Access Table (GOBTPAC) in Banner. N - User IDs and PINs will be authenticated in Luminis Platform, and the Banner settings will be ignored.

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
CPTIMEOUTURL	URL relative to the CPBASEURL which will call the CP timeout authentication logic.
	External system `sct` represents the CORBA-based method of SSO, whereas the external system `sctssb` represents the CPIP delivered with 7.0
	Example for `sct`: `/cp/ip/timeout?sys=sct&api=`
	Example for `sctssb`: `/cp/ip/timeout?sys=sctssb`
CSSURL	This specifies the URL for the external cascading style sheet file used to implement the new user interface. This may be a fully qualified URL, or be relative to your web server document root directory. The default is `/css/twbhstyl.css`
EPOSPAYMENTSERVER (Obsolete)	This parameter is obsolete as of the 8.3 release.
	The name of the EPOS payment server.
	Example: `http://maldevl6.sct.com/scripts/ips.dll`
ESS_TO_EFC_URL	The URL for the link from the Employee menu in Employee Self-Service to Effort Certification.
HEADERDISP	For Banner implementations that have been set up for Multi-Institution Functionality (MIF). Select this parameter to display the institution identifier in the header of Banner Self-Service pages.
	For more information, please refer to the "Institution Identifier" topic in Chapter 2, "Technical Basics."
HELP_HEIGHT	This parameter allows the user to override the default height 500 of the help window.
HELP-WIDTH	This parameter allows the user to override the default width of 450 of the help window.
HELPURL (Obsolete)	This specifies the URL for the global help file delivered with Web Tailor 1.0.2. This may be a fully qualified URL, or be relative to your web server document root directory. The default is `/wtlhelp/twbhhelp.htm`
	This parameter is no longer used by Web Tailor.
HOMEPAGE_BANNER_URL	This specifies the URL to an image which is shown on non-secured home page.

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
IDMCOOKIE IDMCOOKIEDOMAIN IDMCOOKIEPATH IDMHEADER IDMLOGINURI IDMSSO IDMTIMEOUT	These parameters support single sign-on (SSO) setup through Banner Enterprise Identity Services. For information on how to use these parameters, refer to the *Banner Enterprise Identity Services Handbook*. You will use these parameters only if you are using the Banner Enterprise Identity Services product.
IGNORELOGINCASE	If Y, user IDs during login will not be case-sensitive. Lowercase and uppercase characters in user IDs will be treated as equivalent. For more information, see "Login options for user IDs" on page 83.
IPAYMENTSERVER	For web credit card processing. This parameter is only required if your payment gateway is Oracle iPayment11i Server. Do not use this parameter if you are using the TouchNet Payment Gateway or EPOS Payment Server. For more information about web credit card processing, please refer to the *Web Credit Card Payments Handbook*.
LDAPFUNCTION	Specifies which function will be used to perform the LDAP user mapping during authentication. You will use this parameter only if you are using LDAP to authenticate user credentials.
LDAPL4FLAG	For authenticated searches against a Luminis IV LDAP server. Set the value to Y to enable login to SSB using LDAP authentication with Luminis IV. Leave the default value of N if you are not using LDAP with Luminis IV. For more information, see "LDAP authenticated search in Luminis IV" on page 93.
LDAPL4FLTR	For authenticated searches against a Luminis IV LDAP server. This parameter's value indicates the prefix used for searching LDAP. The default value is uid. For more information, see "LDAP authenticated search in Luminis IV" on page 93.

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
LDAPMAPUSER	Specifies where LDAP mapping is defined. This parameter will be used by other products and `twbklogn.p_ValidatePin` to see where to look for LDAP mapping. Valid values are: `DEFAULT` - The SPRIDEN ID is the same as the LDAP ID `EXTERNALUSER` - Treat the LDAP user as an external user defined in GOBTPAC `LDAPUSER` - Treat the LDAP user as an LDAP user defined in GOBTPAC `PROMPT` - Prompt each LDAP user for ID You will use this parameter only if you are using LDAP to authenticate user credentials.
LDAPPWDLENGTH	Specifies the length of the LDAP password. You will use this parameter only if you are using LDAP to authenticate user credentials.
LUMINIS_STU_SECURITY_IND	Determines if the Select Term (`bwlkostm.P_FacSelTerm`) and the Student and Advisee ID Selection (`bwlkoids.P_AdvIDSel` or `bwlkoids.P_FacIDSel`) pages are displayed in future sessions when the Add or Drop Classes page (`bwlkfrad.P_FacAddDropCrse`) is accessed through Luminis. `Y` - The Select Term and the Student and Advisee ID Selection pages are displayed in future sessions when the Add or Drop Classes page is accessed through the Luminis. `N` - The Select Term and the Student and Advisee ID Selection pages are not displayed in future sessions when the Add or Drop Classes page is accessed through Luminis.
MAXSEARCHRESULTS	The maximum number of results returned when a user performs a keyword search. Example: 10

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
ONEVENT_<package>	Activates or disables input and output validation for "on event" HTML, such as `onblur`, `onclick`, and so on. The parameters are named `ONEVENT_<package>`. `<package>` represents the first four characters of a package. For example, ONEVENT_BWCK.
	By default, parameters are set to `IGNORE` and are updated to `REPORT ERRORS` when each application posts their required changes. `IGNORE` disables input and output validation for packages that start with the characters identified with the ONEVENT_xxxx parameter. This same convention can be used for local packages if required. If the parameter is set to a value other than `IGNORE` (for example, `REPORT ERRORS` or NULL), input and output validation is activated for the package.
	The following parameters are supported:
	• ONEVENT_BWCK
	• ONEVENT_BWFK
	• ONEVENT_BWGK
	• ONEVENT_BWLK
	• ONEVENT_BWPW
	• ONEVENT_BWRK
	• ONEVENT_BWSK
	• ONEVENT_EWKW
PAYVEND_TRANS_TIMEOUT	Number of minutes after which the payment vendor's Web site times out if there is no activity.
	An update received from the payment vendor after this time limit is treated as "transaction not found". An error message indicates the current date/time, transaction date/time, and the expiration date/time. The suggested value for this parameter is 15.
PAYVEND_URL	Payment vendor's URL. The user's browser is redirected from Banner Self-Service to this URL to complete a transaction payment. Only one vendor can be active at a time. For example:
	http://server.domain.com:8100/pci/gateway
	If the Self-Service Banner session is initiated with "http", then the payment vendor URL must be also be "http". If the session is started with "https", then the URL must also be "https". If this is not the case, the user will be forced to log in again at the end of the process.

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
PINLENGTH	The maximum length of a PIN.
PINNAME	The label you want to use for the PIN field on your LDAP logon page. You can customize this for your institution. You will use this parameter only if you are using LDAP to authenticate user credentials.
PRINTCSSURL	This specifies the CSS to be used to format web pages for printing unless overridden on a specific page. Example: `/css/web_defaultprint.css`
SELECTNULLDISP	This specifies the value to be displayed in each pull-down menu when no items are selected. Example: `None`
SSL_PASSWORD	Password for the key store
SSL_STORE	Path where the keystore is created. This keystore will contain the necessary certificates. Example: E:/keystore/oas1013.jks.
SSL_TYPE	Key store type. Example: JKS - Java Key Store
SYSTEM_NAME	Value which will represent the browse start point Example: Banner_SSB
TOUCHNETPAYMENTGATEWAY (Obsolete)	This parameter is obsolete as of the 8.3 release. This parameter is required only if your payment gateway is TouchNet Payment Gateway. It should be set to the value `TPG`. If you are using the EPOS Payment Server you can disregard this entry. For more information about web credit card processing, please refer to the *Web Credit Card Payments Handbook*.
UI_MINIFY	Determines if minified JavaScript files are used. By default, the parameter is set to use the minified version of the JavaScript files. For more information, see "Minified and Full JavaScript files" on page 107. The following values can be used: `Y` or if the parameter does not exist - minified JavaScript files are used by Cascade. `N` - full versions are used by Cascade.

TWGBPARM_PARAM_NAME	TWGBPARM_PARAM_VALUE (Example/Description)
UI_THEME	This parameter allows you to select a user interface theme. The following themes are available: • cascade • default
USERIDLENGTH	Specifies the length of the LDAP user ID.
USERIDNAME	The label you want to use for the user ID field on your LDAP logon page. You can customize this for your institution.
WEBCTINUSE	This specifies whether WebCT is implemented and in use at the institution. Valid values are: `Y` – the institution has implemented and is using WebCT `N` - no WebCT implementation (the default)

WEBCTLOGIN	This indicates the URL where the installation's instance of WebCT has been installed.
	
	Example:
	`http://webctserver:port/webct/`
	`homearea/homearea`
	
	where `webctserver:port` would be replaced with the correct value for the institution.

WEBUSER	This parameter's value is the Oracle user name that Banner Self-Service will use to connect to Banner's Oracle database. See the related `AUDITUSERID` parameter. The value delivered with this parameter is `UPDATE ME`. You must change this value to the appropriate Oracle ID. Example values are `WWW_USER` and `OAS_PUBLIC`. If this parameter's value is not set, Banner's FGAC functionality (VBS and PII) will not work in Self-Service.

Web Key Parameters Table (TWGRWPRM)

In normal web page processing, parameter values are passed between the web pages and the database packaged procedures. This works well for pages that handle address updates or the display of grades. However, there are many pages which share the same key parameter, such as registration term, financial aid year, or CRN.

Typically, when a user navigates to a page that requires a key parameter, the web system displays the parameter selection page first. For example, if a student attempts to register for a class, he or she will first have to select a registration term from the Select Term page. Once a valid term is selected, the Add/Drop Courses page will appear. If the student wants to view his or her schedule, there should be no need to reselect a term, because one has already been chosen.

To make this possible, the key parameter is stored in the Web Key Parameters Table (TWGRWPRM). Because the table is used only for background processing in the Self-Service products, a form is not associated with it. Whenever a key parameter (such as registration term) is selected, it is stored in TWGRWPRM with the user's PIDM. Each time the user goes to a page that requires a key parameter, the table is checked. If a value is found, it is used; otherwise, the user will be shown the parameter selection page to make a choice.   If the user wants to change the value he or she had selected, the parameter selection page can be chosen from the appropriate web menu page. Selected parameter values are used until the user changes his or her selection, or logs out.

The TWGRWPRM table maintains itself by updating rows whenever possible, rather than inserting new records for each value selected. All existing records for a PIDM are deleted when the user logs out, and again at the next login (in case the Web user forgot to log out correctly).

For more information on the different parameters, refer to "Web Tailor Tables" on page 108

Graphic elements

Graphic Elements lets you create a graphic element, or modify an existing one. To create or modify a graphic element, select **Graphic Elements** from the Web Tailor menu. Click Create or choose an item from the list by clicking its **Image Name**.

> **Note:** The list order is controlled by **Display Descriptive Names** on the Web Rules page. If checked, then the list is ordered by **Image Description**; if unchecked, it is ordered by **Image Name**.

You can optionally do a text search for a graphic element, and then select from the list of search results. You can search by the image's **Name**, or by its **URL**.

To search for and select a graphic element:

1. Type the full or partial name in the **Search by Name** field, or the full or partial file name or URL in the **Search by URL** field. Search text is not case-sensitive.

   > **Note:** If you enter search terms in both fields, **Search by Name** will take priority and the terms in **Search by URL** will be ignored.

2. You can use % as a wildcard character in either field, or _ as a single-character wildcard.

3. Select **Search**.

4. Select the graphic element from the list of search results, or search again.

Web Page Fields

Item	Description
Graphic Element Name	The name of the element you are creating or modifying. Required.
Image URL	The location of the element, as a URL. Required.
Description	The description of the image.
Comments	Enter any comments about the image.
Image Width in pixels	Enter the image's width.
Image Height in pixels	Enter the image's height.
Highlighted Image URL	The URL of the image to switch to when the user selects or moves the mouse over the graphic.
Alternative Text	For user agents that cannot display images, this value specifies alternate text.
Status Bar Text	Enter the text that should appear on the status bar when the user's cursor rests on the graphic. This text will only appear if the image is being used as a URL.
Image Align	Enter the position of the image with respect to its context.
Image Border in pixels	Enter the size of the border that should appear around the graphic.
Vertical Spacing in pixels	Enter the amount of white space to be inserted above and below an image.
Horizontal Spacing in pixels	Enter the amount of white space to be inserted to the left and right of an image.

Login return location

Login return locations define the page that will appear automatically when the user logs on, depending on where they were when they timed out. Login return locations can be used to:

- Create a direct link to a secure Self-Service page from another site, where the user must then login to Banner Self-Service before they can be taken to the page they want to access.

For example, an e-mail to employees about their paycheck stub may include a URL to view their specific paycheck stub. After logging on the user should be directed to the paycheck stub page and not have to navigate through the SSB menus.

- Identify, by page or module, where the user is taken when they must log back in because their session had timed out. If, for example, the user is in Student Self-Service and the session times out, the user is usually returned to the home page. You can, however, set up a login return location for the Student Main Menu; when anyone times out while using Student Self-Service, they will be taken to the Student Main Menu instead.

Web Page Fields

Item	Description
Return Code	Characters that identify the return location.
	For ease and consistency of use in a URL parameter, the only valid characters for the return code are A-Z, az, 0-9, and the special characters $-_.!*'(), . These values will be edited upon creation or modification so that existing values will still work. However, if you are using any additional characters, they should be converted when possible.
Description	The description that corresponds to the code.
Return Location	The package.procedure or menu name where the user will return after they log back in after timing out.
Location is	Identifies if the return location is a procedure or a menu.
Add New Package Associations	The package.procedures identified with this return location code. You can enter others, as necessary.
Add New Module Associations	The modules for which this return location code is in effect.
	Choose module codes from the pull-down menus to add them to this return location code. You can add up to three module codes at one time.

Web Tailor overrides

Banner Self-Service has been designed to provide you with a great deal of control over your institution's Web pages.

Banner Web Tailor was originally designed to be used as a reusable module for generating and controlling HTML presentation interfaces. While primarily designed to

implement the Self-Service products, it was also used as the underlying presentation manager for other products.

Because of this flexibility, Web Tailor will allow you to replace programming logic stored in package/procedure combinations with your own logic.

Select **Web Tailor Overrides** from the Web Tailor Menu.

Enter the package/procedure combination as the **Override Condition**, then enter your package/procedure (in package.procedure format) as the **Replacement Value**.

Programming logic in the Self-Service applications will first check an "override" table to see if an alternate API (procedure/function) is defined. If so, then that code would be run via dynamic SQL. Otherwise, the standard Web Tailor logic is run.

The packages/procedures you can override are:

- `twbkslib.f_centurypivot`

- `twbkslib.f_dateinputfmt`

- `twbkslib.f_fetchpidm`

- `twbkslib.f_fetchpinformatind`

- `twbkslib.f_formatname`

- `twbkslib.f_proceedafterlogin`

- `twbkslib.f_userisdead`

- `twbkslib.p_create_gobtpac`

- `twbkslib.p_fetchroles`

- `twbkslib.p_update_gobtpac`

> **Note:** The TWBKSLIB package consists of a series of utility procedures and functions for dealing with session state, user identity, etc.

LDAP administration

This web page contains your LDAP settings. These settings override the settings on the Enterprise PIN Preferences Form (GUAPPRF) in Banner General.

You can use the LDAP authentication process to authenticate all your users' IDs and passwords. They can use their LDAP user IDs and passwords to logon to the Self-Service applications they need to use. The mapping between the LDAP user ID and the Self-Service user ID can be stored on the LDAP server as an attribute, or it can be stored on the Third Party Access Table (GOBTPAC) in Banner General.

If the user starts by logging on to Banner Self-Service, the page displayed is very similar to the existing logon page, but the **Forgot PIN?** button does not appear. That feature is only available if you use the GOBTPAC table to perform user authentications.

Since your institution may use different terminology for your authentication process, Web Tailor allows you to create custom labels for the user ID and PIN on the LDAP logon page. You can also specify the descriptions and the length of these items. For example, you may want the LDAP PIN to be longer than the six-character Self-Service PIN. You can specify that on the Web Tailor Parameters page - no coding changes are necessary.

Web Page Fields

Item	Description
LDAP Protocol	Select a protocol used for Self-Service authentication from the pull-down menu. Valid values are: `LDAP` - LDAP authentication without Secured Socket Layer (SSL) `LDAP_S` - LDAP authentication with SSL `NONE` - authentication is performed via the GOBTPAC table in Banner General Required.
Search Indicator	Specifies if an anonymous search will be performed to locate the DN entry. Valid values are: `Selected` - an anonymous search will be performed `Cleared` - an anonymous search will not be performed (default) Clear this indicator if anonymous search is not enabled on your LDAP server.
LDAP Server Name	Enter the LDAP server name.
LDAP Port	Enter the LDAP port number.
Search Base for LDAP	Specify where on the LDAP server you want the search for the DN entry to begin. This will reduce the response time for the authentication.
Suffix for LDAP User	Enter the suffix that will be part of the LDAP user ID for all your users. The true LDAP user ID consists of the Search Base, prefix, suffix, and individual user ID.
Prefix for LDAP User	Enter the prefix that will be part of the LDAP user ID for all your users. The true LDAP user ID consists of the Search Base, prefix, suffix, and individual user ID.

Item	Description
Attribute for Banner in LDAP	Enter the name of the LDAP attribute that stores Self-Service IDs.
SSL Wallet Location	Enter the directory and folder that will contain the SSL wallet information.
SSL Wallet Password	Enter the password for the SSL wallet.
SSL Authentication Mode	This specifies the SSL authentication mode. Valid values are: 1 - No authentication is required 2 - One-way authentication is required 3 - Two-way authentication is required

PINs

This chapter includes the following topics:

Secure logins

The information stored in the Banner™ database and presented to the users in the Self-Service products is highly confidential. Therefore, users must log into the Secure Area to view and update it.

To gain access to the Secure Area, users must enter a user ID and a PIN (Personal Identification Number).

The programming logic that controls user IDs and PINs for all the Self-Service products exists in Banner General and Banner Web Tailor.

> **Note:** The programming logic in Web Tailor that authenticates GOBTPAC user credentials is bypassed if your institution uses LDAP to authenticate Banner Self-Service. For more information about LDAP, see "Technical Reference" on page 86.
>
> For processes that synchronize data with Luminis® and WebCT, the current Banner PIN (GOBTPAC_PIN) is supplied as the Password/PIN value.

All the user IDs and PINs are stored in the GOBTPAC table in Banner General and can be maintained using the Third Party Access Form (GOATPAC) or the Third Party Access Audit Form (GOATPAD). Web Tailor uses this information to grant the appropriate access to authorized users.

> ⚠ **Warning!** *The major difference between GOATPAC and the Third Party Access Audit Form (GOATPAD) is the security assigned to the two forms. GOATPAC is available to various administrative users. GOATPAD should be restricted to key security administrators.*

PINs are managed based on a number of institutional PIN control rules, which are displayed and maintained on the Enterprise PIN Preferences Form (GUAPPRF) in Banner General. For more information, see Banner Online Help and the *Banner General User Guide*.

PIN encryption

Beginning with Release 8.0, PINs are stored in the database in encrypted form. A cryptographic hash is used to transform the PIN before it is stored, so that the unencrypted form of the PIN can never be retrieved from the database. See the "Web Management" chapter of the *Banner General User Guide* for more information on PIN encryption.

Assigning PINs

In Banner, PIN values can be created in a number of ways, and can be updated by authorized administrative users or the Self-Service users themselves. There are three distinct ways in which PINs can be created by authorized administrative users:

- PINs can be assigned one user at a time, using one of two online forms: the Third Party Access Form (GOATPAC) and the Third Party Access Audit Form (GOATPAD). See "Managing User PINs" in the "Web Management" chapter of the *Banner General User Guide* for specific instructions.

- PINs can be assigned for a group of persons simultaneously, using the PIN Creation Process (GURTPAC). See the "Reports and Processes" chapter of the *Banner General User Guide* for information on the GURTPAC process.

- PINs can be assigned, at institutional option, by database triggers on the General Person Base Table (SGBSTDN), Faculty Member Base Table (SIBINST), or the Financial Aid Applicant Status Table (RORSTAT). See below for details.

    Note: If your institution is using an LDAP server to authenticate user logons, you cannot assign PINs in Banner General. They must be assigned in LDAP. Web Tailor's GOBTPAC user credential authentication is bypassed if your institution uses LDAP to authenticate Banner Self-Service.

Assigning PINs via database triggers

Data synchronization currently focuses on students and instructors. (Employees and alumni can also be synchronized with the Luminis Platform alone.) Outside of synchronization processing with partner systems, PINs are also required for access to the other Self-Service applications and Voice Response. Because a PIN is required to access all of these Self-Service applications, you may want to create PINs based upon standard operating practices when the end-user first requests access.

Automatic PIN generation can be accomplished by implementing database triggers processing on three source tables associated with student-related processing.

The three tables are:

Role Assignment	Source Table
Student:	SGBSTDN
Instructor:	SIBINST
Financial Aid:	RORSTAT

When trigger processing is implemented for a table, a database trigger fires when a record is inserted into a source table. The triggers call a standard PIN generation function, which will create a PIN record if one does not exist, using the rules maintained on the Enterprise PIN Preferences Form (GUAPPRF). Because the standard PIN processing function is called by the triggers, all appropriate associated records (GOBPTAC, GOBSRID, AND GORPAUD) are also generated.

Rules for each source table control whether PIN trigger creation is implemented. These rules are maintained on the Enterprise PIN Preferences Form (GUAPPRF). One row displays in the Table PIN Creation Controls Block for each source table that has associated PIN creation trigger processing.   The PIN creation trigger fires on insert of a new record into each table when **Create PIN?** is set to Y.

> **Note:** As delivered, the **Create PIN?** value is Y for all three tables. If you don't want to use it on GUAPPRF, then you must manually set the **Create PIN?** to N so the GOBSRID, GOBTPAC, and GORPAUD tables will not be automatically updated.

PIN creation controls can also be maintained in the Crosswalk Validation Form (GTVSDAX). In GTVSDAX, the PIN creation controls are identified by the Internal Group Code CREATEPIN. PIN creation controls are displayed and maintained on the GUAPPRF so that all PIN controls can be displayed and maintained on one form.

The Enterprise PIN Preferences Form (GUAPPRF) is described in greater detail in Banner Online Help and the *Banner General User Guide*.

Resetting PINs

An administrator can reset a user's PIN in the GOATPAD form. See "Managing User PINs" in the "Web Management" chapter of the *Banner General User Guide* for specific instructions.

The Third Party Access Audit Form (GOATPAD) also displays a history of PIN changes.

Enterprise PIN preferences

PIN processing in Banner is controlled by a number of PIN rules that are displayed and maintained on the Enterprise PIN Preferences Form (GUAPPRF). Values maintained on this form control the format of generated PINs, and whether PINs are automatically

created based upon insertion of records into source tables. It also controls other items associated with PIN processing.

GUAPPRF's **Maximum PIN Length** setting establishes a maximum length for PIN values at your institution. This setting also affects the size of the **PIN** field on the Self-Service login page.

GUAPPRF is described in Banner Online Help and the "Web Management" chapter of the *Banner General User Guide*.

Note: The PIN characteristics set up on GUAPPRF are ignored when you are using LDAP to authenticate your users.

Warning! *Although GUAPPRF provides an option to reset PINs to the user's birth date, U.S.-based institutions should not use the birth date option. According to the U.S. Family Policy Compliance Office (FPCO), using a student's birth date when assigning PINs is a violation of the Family Educational Rights and Policy Act (FERPA).*

Info Text for Expired PINs

Beginning with Release 8.4, information text is used for messages that are displayed when a user gets a PIN expired message, instead of the hard coded messages.

The hard coded messages for the following messages can now be overridden with info text on `twbkwbis.P_ValLogin`. If no info text is found, then the existing messages will still be displayed.

* EXPIRED Your PIN has expired. Please change it now.

* EXPIRE1 Your PIN will expire in 1 day. You may change it now, otherwise enter your PIN in the Re-enter Old PIN field and click the Login button.

* EXPIREDAYS Your PIN will expire in <**VALUE1**> days. You may change it now, otherwise enter your PIN in the Re-enter Old PIN field and click the Login button.

Note: You must use <**VALUE1**> in the location of the message where you want the number of days to appear.

Login options for user IDs

Three Web Tailor parameters control institution-wide options for Self-Service logins.

Log in with Social Security Number

You can set the `ALLOWSSNLOGIN` parameter to `Y` to allow users to enter a Social Security Number (SSN) in place of their ID when logging in.

If the value entered for ID is not found in `SPRIDEN_ID`, the system will check `SPBPERS_SSN` for a unique matching value.

There are some situations where SSN login cannot be used.

- If the SSN value is not unique (if it matches more than one person record), a login using SSN will fail. This means that users who have been assigned a generic SSN value (for example, `999999999`) because their actual SSN is unknown will not be able to use SSN login even if the feature is enabled.

- This feature should not be used at institutions where the values in the SSN field are longer than 9 characters. The `ALLOWSSNLOGIN` option does not increase the size of the login ID field, which remains limited to 9 characters, while the SSN field has been expanded to allow values with up to 15 characters (depending on the institution's SSN preference). Since a user can enter only 9 characters in the login ID field, a comparison with a 10-digit or longer SSN value will fail.

Ignore case of Login ID

You can set the `IGNORELOGINCASE` parameter to `Y` to indicate that lowercase and uppercase letters in User IDs should be treated as equivalent during login. For example, if a user's ID is `ABC12345`, and the user types `abc12345`, the user will get a login error if this parameter is set to `N`, but will log in successfully if this parameter is set to `Y`.

> **Note:** Although a U.S. Social Security Number typically does not include alphabetic characters, the `IGNORELOGINCASE` parameter applies to any alphabetic characters in the SSN field when the SSN is used in place of the user ID for login. In other words, if both `ALLOWSSNLOGIN` and `IGNORELOGINCASE` are enabled, the case of any SSN entered during login will be ignored.

> **Warning!** *With IGNORELOGINCASE enabled, if there happen to be two users (two PIDMs) whose IDs are identical except for case differences, neither user will be able to log in. For example, if User 1 has an ID of `ABC12345` and User 2 has an ID of `abc12345`, then both users will find that they cannot log in. The system will not be able to determine which user is trying to log in and will unable to process the login.*

Capture Banner User ID

The Web Tailor parameter `AUDITUSERID` controls how the user ID is stored with data records changes through Banner's self-service applications.

Banner's self-service products connect to the Oracle database with a proxy ID. This is the ID specified by the WebTailor parameter WEBUSER. In many cases the WEBUSER parameter value is set to WWW_USER or OAS_PUBLIC. The self-service proxy ID is the same for all users of the self-service products.

The default value of the AUDITUSERID parameter is WEBUSER. With this value, when Banner data is changed through Self-Service, the self-service proxy ID is the only identifier stored in Banner tables as the user ID associated with the updated record.

If you need to audit Banner data and find out which Banner user made a specific change to the data, you must set the AUDITUSERID parameter to a value of BANNERID. Then the Self-Service user's Banner ID will be captured by the function twbklogn.f_get_banner_id according to the rules below:

1. If the user is logged in with an Oracle ID that is not the WEBUSER-specified ID, then that ID is captured and preceded by W:—for example, W:SAISUSR.

2. If the user is logged in as the WEBUSER-specified ID, and the user has a GOBEACC record, then the user's logon Oracle ID is captured (again preceded by W:).

3. If there is no GOBEACC record, the user's SPRIDEN ID is captured (again preceded by W:).

4. If no SPRIDEN ID exists, then the WEBUSER-specified ID is captured (again preceded by W:).

In any case, the W: indicates that the record was updated by a Self-Service user. A WS: prefix indicates that the user was a non-secured Student user.

> **Note:** The AUDITUSERID parameter is not implemented in all Banner Self-Service products. Check the documentation for each product to find out if this feature has been implemented.

Technical Reference

This chapter includes the following technical topics:

Baseline and local records

Menus, menu items, roles, and Info Text are separated into two categories:

1. `Baseline` - you cannot change baseline records in any way.

2. `Local` - you can create local records and modify them, and they will never be overwritten by later versions of Banner. You can customize menus, menu items, roles, and Info Text as much as you want without losing anything during upgrades.

When Banner™ attempts to display a record, it first checks for a local record. If it cannot find one, it looks for a baseline record.

The TWGBWMNU, TWGRWMRL, TWGRMENU, and TWGRINFO tables have a source indicator field. Baseline rows contain a `B` and local rows contain `L`. All existing rows are delivered with `B` in this field.

It is not possible to display a combination of baseline and local rows. Each package/ procedure combination will show all baseline rows or all local rows.

When you copy items from baseline to local, Banner will only display the local rows for that menu or procedure. If you then delete any of the local rows, the baseline version will not reappear. The only way to redisplay the original baseline rows is to delete all of the local

ones for that package/procedure combination. As long as there is a local row for that package/procedure, no baseline rows are displayed.

If you create a local version of a menu, you will need to check each upgrade to see if there is a change to the baseline version of that menu. If a new item has been added to the baseline version, you will need to add it manually to your menu. It is not added automatically.

Likewise, if the items on a baseline menu are reordered and you want that order, you must change the order on your local menu manually. The order will not be changed automatically.

Web menus, procedures, and associated roles

You can maintain Web menus and procedures, and their associated roles from Web Menus and Procedures on the Web Tailor Administration menu. They are stored in the TWGBWMNU and TWGRWMRL tables.

Select the menu or procedure from the pull-down list, then choose **Customize Web Menu or Procedure**. You will see the attributes for the menu or procedure, and its associated roles.

Select **Copy Baseline to Local** to make local copies of all the baseline records. Both the baseline records and the local copies are displayed. You cannot edit the baseline records.

Change the local records as necessary, then select **Submit Changes**. To delete a menu or procedure, select **Delete this Local Entry**.

Menu items

You can maintain the items that will appear on menus from **Menu Items** on the Web Tailor Administration menu. They are stored in the TWGRMENU table.

Select the menu or procedure from the pull-down list, then select **Customize Menu Items**. You will see the menu items for the menu or procedure.

Select **Copy Baseline to Local** to make local copies of all the baseline records. Both the baseline records and the local copies are displayed. You cannot edit the baseline records.

To add menu items, select **Reorder or Customize Menu Item**, then **Add New Menu Item**. A page appears that allows you to enter the new item. If you choose a sequence number that is lower than an existing sequence number, the new element will be placed in the sequence you chose, and the other elements will be moved down on the menu.

To delete a menu item, select **Reorder or Customize Menu Item**. Select the item by clicking on its URL. The Customize the Selected Menu Item page appears. Select **Delete this Local Entry**.

When there are multiple items on the menu, the sequence numbers become select lists. You can change the order of the items by choosing appropriate numbers from the select lists. When finished, select **Reorder these Elements**. The page appears with the elements in the new order.

Information Text (Info Text)

You can maintain Info Text by selecting **Information Text** on the Web Tailor Administration menu. It is stored in the TWGRINFO table.

Select the appropriate menu or procedure from the pull-down list, then choose **Reorder or Customize Information Text**. You will see the Info Text for the menu or procedure.

Select **Copy Baseline to Local** to make local copies of all the baseline records. Both the baseline records and the local copies are displayed. You cannot edit the baseline records. The labels of the local entries are links, which you can select to modify the Info Text item. The system will only copy baseline records with no corresponding local record; no existing local records can be modified by this option.

To add Info Text, select **Reorder or Customize Information Text**, then **Add New Information Text Entry**. A page appears that allows you to enter the new item. If you choose a sequence number that is lower than an existing sequence number, the new element will be placed in the sequence you chose, and the other elements will be moved down on the menu.

To delete an Info Text entry, select **Reorder or Customize Information Text**. Select the item by clicking on the link on the label. The Customize the Selected Information Text Entry page appears. Select **Delete this Entry**.

When there are multiple Info Text items for a label, the sequence numbers become select lists. You can change the order of the items by choosing appropriate numbers from the select lists. When finished, select **Reorder these Elements**. The page appears with the elements in the new order.

> **Note:** It is recommended that you use the existing procedure `twbkwbis.P_DispInfo` to retrieve and display Info Text on new pages you create, or existing pages that you modify to display Info Text. Please refer to the Web General packages for specific examples of the `P_DispInfo` procedure.

Help Info Text

A system of help messages stored has been stored as Info Text with the label `HELP`. It was originally stored in the Banner Self-Service products in static HTML files. By changing it into Info Text, it will be much more easy for non-technical users to customize and maintain.

> **Note:** The static HTML help system will continue to be supported. It was modified to conform to the same accessibility standards as Info Text.

As with all other rows, Info Text Help records are stored in Web Tailor as baseline. Info Text rows created by your institution using Web Tailor are stored as local. You cannot modify baseline rows. Info Text that is stored as local is displayed first. If there are no rows stored as local, the baseline rows are displayed. This ensures that your institution's custom help will not be overwritten with each Banner upgrade.

> **Note:** Web Tailor still has help messages stored in static HTML files (help text that describes them). This help is displayed when the user selects the **Help** link in the upper right-hand corner of the Web page. The help text appears in a new window, and includes an **Exit** link at the bottom of each window.

A CSS for global-level help, `web_defaulthelp.css`, is provided. It will be used in addition to the CSS defined for that page.

Internationalization

The Banner Self-Service products contain logic to facilitate translating text to targeted languages, and to support other date and number formats.

Appropriate hard-coded strings in the Web packages have been replaced by calls to the package called `G$_NLS` so that they can be translated into various languages. These changes are transparent to the user, and do not affect clients using American English.

The schema owner NLSUSER owns the package `G$_NLS` and the version table TMURVERS. A record has been added to the GENERAL.GTVSYSI table, with `GTVSYSI_CODE=TM` and `GTVSYSI_DESC=Translation Manger`.

> **Note:** If you need to implement multilingual support for your Banner Self-Service products, it is recommended that you contact Professional Services to help you set up this feature.

Web session processing and security

There are two important security considerations to address with respect to allowing end users access to the Banner administrative database through the Web:

- You must ensure that any attempt to break into the system with existing Banner IDs (e.g., by guessing PINs) is tracked and Web account access is disabled if appropriate.

- You must track Banner database access and ensure that the ID currently logged on is accessing and updating only records associated with that ID.

The Web Session Information Table (TWGBWSES) supports session security for both of these concerns.

Preventing break-ins

To address the first situation, a counter for tracking login attempts associated with an existing ID is updated in the Web Session Information table (TWGBWSES).

The counter (`TWGBWSES_LOGIN_ATTEMPTS`) has an initial value of zero. If the ID is valid but the PIN is not, the value of the counter is incremented by one. Additional attempts for the same ID are permitted, up to the value specified for the number of Login Attempts in Web Tailor. When the counter reaches the value specified for Login Attempts, the account is disabled, and the counter is reset to zero.

There is always a possibility that unauthorized individuals may obtain and use the valid ID and PIN of a user. To help someone detect unauthorized use of his or her ID and PIN, an information message displays the last access date and time on the Main Menu after login. This information comes from the Last Web Access date stored in the TWGBWSES table.

The date field (`TWGBWSES_LAST_ACCESS`) stores both the date and time the ID last accessed the Web (initial login date and time of the session, not the logout date and time). It may be helpful to remind users that if the last login date and time is suspicious (that is, they did not access the system at that time), they should suspect fraudulent use of their ID and PIN, and report it to the system administrator.

Monitoring ID/PIN instances

To address the second concern, Banner Self-Service uses hidden and randomly-generated Web IDs during the entire Banner Self-Service session to track whether the ID and PIN logged on are, in fact, the same ID and PIN for the duration of the session. The `TWGBWSES_WEBID` field and the `TWGBWSES_PREV_WEBID` fields track this information.

A function to generate a random WEBID initially populates the `TWGBWSES_WEBID` field after successful login. At the start of each new database access, the current value of `TWGBWSES_WEBID` is moved to `TWGBWSES_PREV_ID`, and a new randomly-generated Self-Service ID populates `TWBKWSES_ID`.

With each access, a check is performed, comparing the value of the Web ID on the Web page with the value stored in the database. If they are the same, the user can continue. If they are not the same, it is assumed that a break-in has been attempted, and a message will be displayed requiring the user to login again.

If Self-Service session inactivity has caused a timeout (no database access for the number of minutes specified in `TWGBWRUL_TIME_OUT`), `TWGBWSES_WEBID` and `TWGBWSES_PREV_WEBID` are set to null. The system displays a message indicating that a timeout has occurred and that the user must log in again.

Both fields are also set to null when the user logs out of the Self-Service session.

Tracking a current Self-Service ID and a previous Self-Service ID accomplishes two security goals:

1. It prevents a user from running two sessions simultaneously for the same ID and PIN, as the updates to TWGBWSES will conflict with each other. One of the sessions will receive the break-in attempt message.

2. It prevents a completely different user from copying the URL information and attempting to enter the URL in an effort to illegally gain access to the person's information.

Because most of the fields in the TWGBWSES table are for background processing in the Self-Service products, there are no online forms associated with the table, with the exception of one of the fields.

A single row of data is created in the TWGBWSES table when an end user first logs on to the system. This row, which is keyed by the Banner PIDM associated with the end user, is never deleted, but is updated as described above. The row is not deleted when the user logs out, the next login will determine that a row for the PIDM exists in the table, and updates will be performed as described.

Using LDAP for user authentications

Overview

You can use the LDAP authentication process to authenticate all your users' IDs and passwords. They can use their LDAP user IDs and passwords to logon to the Self-Service applications they need to use. The mapping between the LDAP user ID and the Self-Service user ID can be stored on the LDAP server as an attribute, or it can be stored on the Third Party Access Table (GOBTPAC) in Banner General.

Note: The programming logic in Web Tailor that authenticates user credentials in GOBTPAC is bypassed if your institution uses LDAP to authenticate Banner Self-Service.

If your institution is using an LDAP server to authenticate user logons, you cannot modify PINs in Banner General. They must be changed in LDAP.

Authentication process

The authentication process is as follows:

1. The Web Tailor package TWBKLOGN checks to see what protocol you have chosen to use, that is, what you selected from the **LDAP Protocol** pull-down menu on the new LDAP Administration page in Web Tailor:

   • If you have selected LDAP, you will use generic LDAP to authenticate your users

   • If you have selected LDAP_S, you will use LDAP with Secured Socket Layer (SSL).

- If you selected `NONE`, you will use the values in the GOBTPAC table for authentication

2. When a user first tries to logon, TWBKLOGN checks to see if the **Search** indicator was selected (`TWGBLDAP_SEARCH_IND = Y`) on the new LDAP Administration page in Web Tailor.

    2.1. If this option is selected, an anonymous search will be performed to see if the user exists in LDAP, or if the user exists in more than one place. LDAP will use the Distinguished Name (DN) found in the search in the bind.

    If either of these conditions generate an exception, TWBKLOGN displays an error message and does not allow the user to logon. The error message is Info Text, and you can customize it in Web Tailor.

    2.2. If this option is not selected, the bind will be performed directly using the parameters you specified on the LDAP Administration page in Web Tailor.

3. Banner Self-Service provides two ways of mapping LDAP users to Self-Service IDs:

    3.1. The LDAP user can be mapped to Banner Self-Service by storing Self-Service IDs as an attribute in LDAP.

    3.2. The LDAP user can be stored in the `GOBTPAC_LDAP_USER` column of the GOBTPAC table in Banner General.

When you logon for the first time, you are presented with a Terms of Usage page. This feature has not been changed for the LDAP enhancement.

For additional technical information, please refer to the *Banner Middle Tier Implementation Guide.*

Prerequisites

If you are planning to use LDAP to authenticate your users' credentials, you must also have the following Oracle packages in your LDAP directory.

> **Note:** The creation scripts can be found in `<ORACLE HOME>/rdbms/ admin`. When you run them, you should be connected as sysdba.

Package Name	Creation Scripts

`dbms_pipe`	`dbmspipe.sql`
	`prvtpipe.plb`
	
	Run these scripts as part of `catproc.sql` during a standard Oracle install.

`dbms_random`	`dbmsrand.sql` Run this script as part of `catproc.sql` during a standard Oracle install.

Package Name	Creation Scripts

`dbms_ldap` and `dbms_ldap_utl`	`prvtldh.plb`
	`dbmsldap.sql`
	`prvtldap.plb`
	
	These are run in the proper sequence in the `catproc.sql` script.

`dbms_obfuscation_too lkit`	`dbmsobtk.sql`
	`prvtobtk.plb`
	
	These are run in the proper sequence in the `catproc.sql` script.

`utl_raw`	`utlraw.sql`
	`prvtrawb.plb`
	
	These scripts are run as part of `catalog.sql` during a standard Oracle install.

`utl_encode`	`utlenc.sql`
	`prvtenc.plb`
	
	These scripts are run as part of `catalog.sql` during a standard Oracle install.

`utl_url`	`utlurl.sql`
	`prvturl.plb`
	
	These scripts are run as part of `catalog.sql` during a standard Oracle install.

WEB PL/SQL Toolkit `htp`, `htf`, `owa`, `owa_util`	`owainst.sql` This script conditionally runs the `owacomm.sql` package. To force an install, run `owacomm.sql`.

LDAP authenticated search in Luminis IV

You can set up Banner Self-Service to do authenticated searches against a Luminis IV LDAP directory.

1. In Web Tailor's LDAP Administration page, enable the Search Indicator.

2. In the Web Tailor Parameters page, set these parameters:

• Set LDAPL4FLAG to `Y`.

• Set LDAP4FLTER to `uid`. This is the prefix used for searching LDAP.

Self-Service and FGAC

Banner's Value-Based Security (VBS) uses the Oracle FGAC tool, and is applicable throughout all of Banner. VBS restricts the access of your end users to the database, regardless of how they accessed it. Their access depends upon institution-defined rules that control what data they can select, update, and delete.

You can implement VBS with FGAC for all Banner processing, including the Self-Service products.

When you access Banner Self-Service and VBS restrictions exist for your role in Banner (e.g., `Faculty`), nothing will appear different to you until you try to access a Web page that has been restricted on the Customize a Web Menu or Procedure page (twbkwmnu.P_ModifyPgWebMain). You will then be asked to enter your Oracle username and password. All the VBS restrictions for your role will be applied to your access during your session. You do not have to enter your Oracle username and password again, even if you try to access another Web page that has been restricted.

Note: It is recommended that you do not restrict Web Tailor pages.

If you are accessing Banner Self-Service from Luminis®, the VBS restrictions go into effect automatically when you access a restricted Web page. You do not need to enter your Oracle username and password at that time because the Single Sign-On (SSO) feature handles it automatically.

Warning! For VBS to work properly with Banner Self-Service, any Oracle IDs that will be used with the Self-Service applications must be granted the role `ban_default_webprivs`*. For more information about this role, see the Banner Security Administration Handbook.*

For more information about Value-Based Security, see the *Banner Data Security Handbook.*

WEBUSER parameter

The WEBUSER parameter on the Customize a Web Tailor Parameter page (`twbkparm.P_DispAllParams`) contains the Oracle user that Banner Self-Service will connect as. VBS needs this value to function appropriately.

Warning! The value delivered with this parameter is `UPDATE ME`*. You must change this value to be the Oracle ID your users will use to connect to Banner Self-Service (e.g.,* `OAS_PUBLIC`*).*

The WEBUSER parameter is stored in the Web Tailor Web Parameters Table (TWGBPARM) in Web Tailor, and can be customized via the TWBKPARM package.

Securing web pages

The **Admin Secured** check box on the Customize a Web Menu or Procedure page (`twbkwmnu.P_ModifyPgWebMain`) specifies if that Web page is secured under the VBS and PII rules set up in Banner.

If you select this check box, a user who accesses the page must specify an Oracle username and password, and the VBS and PII rules will apply to the user's activities for the duration of that person's session.

> **Note:** VBS and PII rules are defined for the user's role, e.g., `Faculty`.

Each user who you want to restrict via VBS must have a record in the Enterprise Oracle Access Table (GOBEACC).

Database Access Descriptors (DADs)

The architecture of Banner Self-Service uses Database Access Descriptors (DADs) for database connection information. The connection information can contain a user ID/password combination.Banner Self-Service recognize two different DADs:

1. The DAD with the user ID/password combination is the standard DAD in use before 7.0, and it is used for normal user ID and PIN logins.

2. Another DAD introduced with Release 7.0 does not have a user ID/password combination. It is used when access to a specific Web page should be restricted under the VBS rules set up in Banner. When the new DAD is being used, Banner Self-Service will prompt each user for his or her Oracle user ID and password to gain access to that page. The user will only need to supply the ID and password once each session, and VBS will be in effect for that user for the remainder of that person's session.

   This DAD is used for all transactions when the **Admin Secured** checkbox is selected on the Customize a Web Menu or Procedure page (`twbkwrul.P_ModifyPgWebRules`).

   > **Note:** After you successfully access a Web page that has been secured via the **Admin Secured** checkbox, that level of access is what you will have throughout the session.

Browsers and caching

Browsers can store copies of recently accessed pages on local computers so the page does not have to transmitted across the network the next time you access it.

While this is helpful for performance, it does create a security issue for educational institutions where computers are often shared in public settings like labs and libraries.

If your browser has cached a recently-visited page with sensitive information, it may be possible for the next user to retrieve that page from the cache.

The issues are:

- How to control which pages are cached by the browser

- What controls are available when someone accesses a page via the **Back** button

To control which pages are cached by the browser, you should make sure the following are always set:

- For Netscape:
  - Find Edit>Preferences>Cache>Compare this page in the cache to the next page on the network
  - Select `Everytime`

- For Internet Explorer:
  - Find Tools>Internet Options>Advanced>Security
  - Select `Do not save encrypted pages to disk`

To control what happens when someone uses the **Back** button is more difficult. We have found that different versions of the same browser from the same company can ignore or comply with HTML caching directives when re-displaying a cached page via the **Back** button.

> **Note:** The best way to make data secure is to train users to close the browser after they log off, as directed on the Logout page.

Packages and procedures

Procedures

A procedure (also referred to as package.procedure) is code (PL/SQL or C) that executes to carry out a specific function on the Web. The procedures that are delivered in the TWGBWMNU table include those which create and display static Web pages, such as the college catalog, class schedule, and campus directory, as well as those which generate dynamic Web pages.

You can group several functionally-related procedures together in one package. The overall package contains different sections of code for each procedure within the package. For example, the process of viewing address information on the Web belongs to the package that includes all address functionality, while the process that changes a PIN belongs to the package that contains other PIN functions such as logins and security checks.

Packages are compiled in the database from two separated files:

1. The *package specification* defines the public interface for the procedures and functions in the package. It has the full eight-character name, for example, `twbkwbis.sql`.

2. The *package body* defines the actual implementation of those procedures and functions. It has the same name as the package body, but the last character of the name is `1`, for example, `twbkwbi1.sql`.

Some functions, such as viewing an address, consist of multiple steps or procedures that the end user performs (selecting the address type to view, then actually viewing the address information). Default values for many of the fields are delivered with installation, and should *not* be modified unless customization is desired.

Modifying existing procedures

It is recommended that you keep a detailed record of any changes that you make to a Self-Service product. It will assist you when subsequent baseline releases are delivered and you need to analyze baseline changes compared to your customizations.

Use extreme care when changing any existing delivered procedures. You should be familiar with PL/SQL and database packages, procedures and functions. Many Self-Service pages pass parameters required for successful execution of the package.procedure. Your changes to an existing page should successfully pass and correctly use any parameters that may be required. Also, you should know that many procedures and functions are called from multiple packages and even different Self-Service products.

Before undertaking any modification you should:

1. Determine the functional need for the modification

   Is the change appropriate for all of the Web roles that are enabled for the page? If not, you may need a new procedure for the specific Web role or roles.

2. Determine the appropriate package.procedure where the modification needs to be made by the package.procedure name displayed in the URL.

   If the package.procedure name is, for example, `twbkabcd`, the package specification will be found in the appropriate Self-Service product/module (see the section on Web naming conventions for more information) in the `dbprocs` subdirectory with the name `twbkabcd.sql`. The specific code, known as the package body, for the individual procedure will be found in the same subdirectory with the name `twbkabc1.sql`.

3. Determine the technical modification required to meet the functional need.

   Complete the code change. Recompile the affected procedures and test.

4. Determine whether changes to links to or from other pages would be appropriate, as well as changes to Info Text. Make changes accordingly.

Please note that small changes can probably be completed easily. These could include a modification to the Change PIN page which would prevent a change to any previously used PIN. Following the guidelines listed above, you would:

1. Determine the functional need for the modification.

Do you want the modification to affect all users, or maybe just faculty and employees? Do you want to prevent the use of any previously used PINs or do you want to restrict it to only those used within the last year? This is the time to make a specification of your changes. For this example, let's assume the change will affect all users and prevent the reuse of any PIN in the PIN Audit Trail (GORPAUD) table.

2. Find the appropriate package.procedure.

    Select the Personal Information Menu in the pull-down menu on the Menu Items page in the Web Tailor. Look for `Change PIN`. The URL field tells you that the package.procedure is `twbkwbis.P_ChangePIN`.

    The package source code can be found in the `dbprocs` subdirectory for the appropriate product.   Except for very small packages, it is a standard that the body of the package will be in a file with the same first seven letters as the package, but it ends in the number 1 (one). So look for a file in `wtlweb/dbprocs` called `twbkwbi1.sql`.   Once in that file, find the procedure `P_ChangePin`. As you look at the code for this procedure, you will notice that another procedure, `P_ValidateChangePin` is called. This is the actual procedure which validates all the requirements for a PIN change. It is separated from `P_ChangePin` so that it can also be called from `P_ValLogin`, in case an expired PIN needs to be changed during the login process.

3. Determine the technical modification.

    In the package goklibs, there is an existing function, `F_PinReused`, that checks for PIN reuse. You could either insert a call to that function, or write one specifically for your institution. You will also need to create an error message and pass it back to the `P_ChangePin` procedure correctly. Remember, this change will also be in effect for any PIN changes which occur during the login process due to PIN expirations.

4. Determine whether changes to links to or from other pages would be appropriate, as well as changes to Info Text.

    Since we did not create a new page or drastically change the function of an existing one, no link changes are required. However, it would be wise to add some Info Text notifying your users that PINs may not be reused.

Adding new menus and procedures

Use Web Tailor to define both the Web menu structure and the appearance of each page. All dynamically-generated menus and interface procedures that are called from the Oracle Apache PL/SQL Agent must be defined. You must make your pages available to a specific role or roles using Web Tailor, but there may be additional steps, depending on the role. For more information, please refer to the product-specific documentation.

Menu pages can be easily generated by calling the existing procedure `twbkwbis.P_GenMenu`; when you use the submenu indicator on the Customize the Selected Menu Item page, that procedure is automatically used.

It is highly recommended that functional pages call `twbkwbis.P_OpenDoc` to generate the HTML document open, title, body, and header tags that begin a page. `P_OpenDoc` uses the information provided in twgbwmnu when it generates this code. The procedure

`twbkwbis.P_CloseDoc` prints any bottom-of-page links associated with the page. `P_OpenDoc` also provides accessibility functionality.

Parameter `print_module_bottom_links` was added to suppress module bottom links for printing. The default value for this parameter is `true`.

All procedures which use `P_OpenDoc` to open the page *must* use `P_CloseDoc` to close the page, and vice versa, due to Web Tailor's use of HTML tables for page formatting purposes.

> **Note:** `P_CloseDoc` cannot be used in a page unless `P_OpenDoc` is also called. Please refer to any of the General Web packages for a specific example of the `P_OpenDoc` and `P_CloseDoc` procedures (for example, `twbkwbi1.sql`).

Web roles

User roles in Self-Service determine what menus are displayed after logging on and what each person has access to. Users can see and access only those menu items that their roles authorize them to see. A user may have more than one role.

> **Note:** Do not confuse the concept of web user roles with the Oracle roles used in Banner security. For information about Banner security's use of roles, see the *Banner Security Administration Handbook.*

Roles are not the only factor that determine which pages a user can access. For example, a student will not be allowed to register for classes if the individual's current general student record is not active.

There are three types of roles that each user may be assigned at the time that they login. Each valid role that is found is loaded into an occurrence in `twbkslib.role_table` during login by the `twbkslib.p_fetch_roles` procedure.

- Standard Banner roles`.` Each of these roles is defined by the existence of data in the Banner system via the GOVROLE view. It is possible for a user to have none, any, or all of the roles.

- The All Web Users role `(WEBUSER).` This role is assigned dynamically to every user that logs in.

- Manually assigned roles. These roles are assigned to a user by a Web Tailor Administrator on the Update User Roles page in Web Tailor. It is possible for a user to have none, any, or all of the roles. The assigned roles are stored in the TWGRROLE table.

Refer to each Self-Service product's documentation for more information on the roles required for its users and the application rules that control a user's access and update privileges.

Standard Banner web roles

Each of these roles is automatically assigned when a user has the required records in the Banner database. The GOVROLE view gathers the information from the Banner database to determine. It is possible for a user to have none, any, or all of these roles.

Web Role	Banner records required to assign this role	Main Menu selections available for this role
Alumni	a Constituent record (APBCONS)	General Functions Menu Alumni Menu
BSAC	GOVROLE_BSAC_IND = Y	*See Banner Student Aid for Canada documentation*
Employee	an Employee record (PEAEMPL)	General Functions Menu Employee Menu
Faculty	a Faculty Information record (SIAINST)	General Functions Menu Faculty and Advisors Menu
Finance	Finance Security Record from FOMPROF with the Web Access Indicator selected and an Enterprise Access Record from GOAEACC, relating the Oracle ID and Banner ID	General Functions Menu Finance Menu
Financial Aid	a Financial Aid application record (ROASTAT)	General Functions Menu Financial Aid Menu
Friend	see product documentation	
Student	a General Student record (SGASTDN)	General Functions Menu Student Menu

WEBUSER role

The All Web Users role (WEBUSER) is assigned dynamically to every user that logs in. This WEBUSER role has no connection to the TWGMPARM parameter of the same name.

Manually assigned web roles

These roles are assigned to a user on the Update User Roles page in Web Tailor. It is possible for a user to have none, any, or all of the roles. The assigned roles are stored in the TWGRROLE table.

Web Role	TWGRROLE_ROLE Value	Main Menu selections available for this role
Advancement Data Tailor	ADVANCEMENTDATATAILOR	Web Tailor Menu (for Alumni Rules only)
Advancement Moves Manager	MOVESMANAGER	General Functions Menu Development Officers Menu
Development Officer	DEVELOPMENTOFFICER	General Functions Menu Development Officers Menu
Effort Certification Administrator	*see product documentation*	
EPAF Administrator	*see product documentation*	
Executive	*see product documentation*	
Faculty Compensation Administrator	*see product documentation*	
Finance Data Tailor	FINANCEDATATAILOR	Web Tailor Menu (for Finance Site Preferences only)
HR Manager	*see product documentation*	
Labor Redistribution Initiator	*see product documentation*	
Master Salary Planner	*see product documentation*	
Travel & Expense Administrator	*see product documentation*	
Travel & Expense Approver	*see product documentation*	
Travel & Expense User	*see product documentation*	
Web for Executives Administrator	*see product documentation*	
Web Tailor Administrator	WTAILORADMIN	General Functions Menu Web Tailor Menu

Assigning the Web Tailor Administrator role

You can run the twadminu.sql script through SQL*Plus to add the Web Tailor Administrator role to an existing Banner ID. This file can be found in the `BAN_HOME/wtlweb/plus/` directory.

Modifying web roles for existing procedures

Web roles are delivered as appropriate for baseline product functionality. You may remove Web roles from pages or add them to pages as you deem appropriate. When you create roles, be sure to save them as local, so future releases do not overwrite them.

Keep the following points in mind, particularly if you want to add Web user roles to pages that currently do not have that role.

Does the page require certain parameters to be passed? Does the Web user role that you are adding have records in the database that can supply the values needed for those parameters? If not, then the page will not work for that Web role. For example, you should not add the EMPLOYEE role to the Student Self-Service Add/Drop Classes page. An EMPLOYEE who does not also have the STUDENT role would not be able to access the page because the correct record (active general student record) does not exist in the database.

You will probably most often remove roles from selected Web pages, particularly in Web General's Personal Information. For example, you may determine that you will permit the roles of EMPLOYEE and FACULTY to update address information, but not STUDENT. As delivered, all three are allowed to update addresses.

For example, to remove the STUDENT Web role from Update Address capability, you would:

1. Determine the page where the address information is updated.

2. For the package.procedure name for the page, you would select that package.procedure in the Web Menus and Procedures page, then scroll down to the Associated Roles section.

3. Delete the STUDENT role record by clearing the check box next to **Student**

4. Select **Save Changes**.

Adding web roles for new procedures

If you add a new procedure, you must assign at least one Web role, or none of the users will be able to access it. You should assign the appropriate roles depending on the purpose of your new procedure, and the Web product it is associated with. Keep in mind the database record requirements as you determine which Web roles are appropriate to add.

Printing text fields and processing date input

To print text fields and process date input in your own custom procedures, go to the Web Tailor Web Rules page. Adjust the settings for **Date Display Format Mask** and **Date Input Format Mask** during unit testing to verify that your procedures are correctly referencing the current display and input formats.

- To print text input fields, use the following PL/SQL code to print a text field called `START_DATE`:

    ```
    twbkfrmt.P_FormDateText ('START_DATE','','');
    ```

    This code produces both the text field and the description of the date input format. The HTML code for this output looks like the following:

    ```
    <INPUT TYPE="text" NAME="START_DATE" SIZE="12"
    MAXLENGTH="10">  (MM/DD/YYYY)
    ```

- To process the date field created, specify the `START_DATE` as an "in" parameter to your procedure. Use the `twbkwbis.F_IsDate` function to determine if the date is valid using the current date input format. After you determine that a date is valid, use the `twbkwbis.F_FmtDate` function to convert the VARCHAR2 input to Oracle data.

```
procedure P_SaveDate (start_date IN VARCHAR2 DEFAULT null) is
  begin
    IF (start_date IS NOT NULL) AND
       (twbkwbis.F_IsDate(start_date, twbklibs.date_input_fmt))
    THEN
      UPDATE date_table
      SET reg_start_date = twbkwbis.F_FmtDate (start_date);
      COMMIT;
    ELSE
      <Do error processing, the date is null or is not valid>
    END IF;
  . . .
  end P_SaveDate;
```

- To display a date on the Web, use `twbklibs.date_display_fmt` to format your date instead of `twbklibs.date_input_fmt`:

    ```
    htp.p(to_char(reg_date, twbklibs.date_display_fmt));
    ```

Breadcrumb validation

The breadcrumb trail in Banner Self-Service is created based on parent-child relationships between pages in the system. This parent-child relationship is defined in Web Tailor through use of the Back Link URL field and in the menu definition structure (twgrmenu table). However, since some pages are used in more than one area, occasionally the bread crumb trail may not accurately represent the path taken to get to the page.

If a back link URL has an invalid parent, or invalid or empty back link URL, the bread crumb trail is generated using the structure defined in the site map, as long as the following two conditions are met:

1. The page is defined as a child to another page in the twgrmenu table.

2. It is a child to only one parent; otherwise, there is no way to determine who the correct parent for the current view is.

   If these conditions are not met, the bread crumb trail will display the following:

```
Home -> "Page Name"
```

where "Page Name" is equal to the value in Header Details (`TWGBWMNU_HEADER`) if available. If Header Details is not available, Page Title (`TWGBWMNU_PAGE_TITLE`) will display, else Description (`TWGBWMNU_DESC`) will display.

> **Note:** This is applicable only for the Cascade theme.

Empty back link URL

Bread crumb trails may no be generated correctly for pages that do not have back link URLs defined in Web Tailor. To determine if there is a back link URL that can be used to construct breadcrumbs based on the usage of the page, execute the following SQL statement:

```
SELECT twgbwmnu_name, twgbwmnu_source_ind FROM twgbwmnu WHERE
twgbwmnu_name = proc_name AND (twgbwmnu_back_url is NULL OR
twgbwmnu_back_url = '');
```

When the query returns one or more rows it indicates that the back link URL is NULL for that page.

Invalid back link URL

An invalid back link URL can either be a page that has not been defined in Web Tailor, a link to a page outside of Self-Service Banner, or a link to a page that doesn't match the hierarchy defined in the site map.

Even if a back link URL has been defined, the bread crumb trail for the page may still not be able to be generated if the URL is not valid.

To determine if there is an invalid back link URL, execute the following SQL statement:

```
SELECT twgbwmnu_back_url

FROM twgbwmnu

WHERE twgbwmnu_name = proc_name

AND twgbwmnu_back_menu_ind = 'Y'

AND twgbwmnu_source_ind =

(SELECT nvl( max(twgbwmnu_source_ind ),'B') FROM twgbwmnu WHERE
twgbwmnu_name = proc_name AND twgbwmnu_source_ind='L') AND
twgbwmnu_back_url IN (SELECT twgrmenu_name FROM twgrmenu WHERE
twgrmenu_url = proc_name);
```

This query returns NULL for pages which have invalid back URLs defined.

Resolving breadcrumb/page issues

You can query to determine pages that do not generate breadcrumbs as follows:

```
(SELECT t.twgbwmnu_name
     FROM twgbwmnu t
    WHERE t.twgbwmnu_back_menu_ind = 'Y'
      AND t.twgbwmnu_source_ind = 'B'
      AND (t.twgbwmnu_back_url NOT IN
       (SELECT tg.twgrmenu_name
         FROM twgrmenu tg
         WHERE tg.twgrmenu_url = t.twgbwmnu_name)))
     MINUS
     (SELECT t.twgbwmnu_name
      FROM twgbwmnu t
    WHERE t.twgbwmnu_back_menu_ind = 'Y'
      AND t.twgbwmnu_source_ind = 'B'
      AND t.twgbwmnu_back_url IN
   (SELECT t2.twgrmenu_name
    FROM twgrmenu t2
   WHERE t2.twgrmenu_enabled = 'Y'
   AND t2.twgrmenu_source_ind = 'B'
   AND 1 = (SELECT COUNT(*) FROM twgrmenu t3
        WHERE t3.twgrmenu_url = t2.twgrmenu_url
          AND t3.twgrmenu_enabled = 'Y'
          AND t3.twgrmenu_source_ind = 'B')));
```

For all pages in the result set, you will need to define a value back link URL in Web Tailor, or the page should be correctly defined as a child to only one parent in the twgrmenu table.

Handling pages with improperly coded HTML (Cascade theme)

In the Cascade theme, if a page does not use proper HTML, the page may not render properly. The HTML issues may include the following:

- HTML content/tags appearing above the <!DOCTYPE> or <HTML> declaration

- Incorrect <!DOCTYPE> declaration

- Improper closure of <FORM> tags

- <FORM> tags not being closed

- <P> tags not being closed

- Improper closure of <TABLE> tags at certain places

- <TABLE> tags not being closed

Identify <!DOCTYPE> issues

When HTML content/tags appear above the <!DOCTYPE> or <HTML> declaration or an incorrect <!DOCTYPE> declaration is used, the user interface will not display correctly. The display issues include the header and footer being displaced. This error needs to be fixed in the HTML by adding the correct <!DOCTYPE> and moving all HTML content inside the <HTML> tag.

The correct <!DOCTYPE> is as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
www.w3.org/TR/html4/loose.dtd">
```

Identify <P>, <FORM>, and <TABLE> tag issues

When the following issues occur with an HTML page, this indicates that a <P>, <FORM>, and <TABLE> tag may have an issue:

- Form buttons are missing from the content pages

- Form Submits are not triggering

- Form Submits have a PL/SQL failure

Resolve HTML issues

To resolve HTML issues in pages, perform the following steps:

1. Identify the page with invalid HTML. The Page Name is included in the URL. For example, bwskalog.P_DispChoicesStu is the page name in the following URL:

```
http://<hostname>:<port>/<DAD>/bwskalog.P_DispChoicesStu?name=xyz"
```

2. If the generated HTML page has a <!DOCTYPE> issue, modify the PL/SQL package to correct the issue and recompile the PL/SQL package.

3. If the generated HTML page has a `<P>`, `<FORM>`, or `<TABLE>` tag issue and renders properly in the Default theme, open the `/js/serviceProperties.js` file and add the package.procedure name to the exception list and save your changes. For example:

```
var CascadeDowngrade =
{
        exceptions: ['bwskalog.P_DispChoicesStu']
}
```

4. Refresh the page to confirm that the issue was resolved.

Minified and Full JavaScript files

Web Tailor uses JavaScript files in minified and full versions for the Cascade theme. A minified JavaScript file is a compressed version of a JavaScript file with audit trails, extra spaces, new lines, and comments removed. The full versions of the JavaScript files have audit trails, extra spaces, new lines, and comments.

Uses for minified and full versions of JavaScript files

The minified versions are used to improve performance by reducing the file size and allows for it to be downloaded to the page faster. The full versions can be used to troubleshoot problems. By default, twbkwbi1.sql uses the minified versions of the .js files.

Identify minified and full versions of JavaScript files

Minified versions have -min appended to the full version name. For example, cascade-controls.js is the full version, and cascade-controls-min.js is the minified version.

Switch between minified and full versions of JavaScript files

Web Tailor uses the minified version of the JavaScript files by default. The UI_MINIFY parameter allows users to switch between the minified and full versions of JavaScript files. This parameter is added to the **Web Tailor Administration** tab and then set to one of the following values:

- When the UI_MINIFY parameter is set to Y or if the parameter does not exist, the minified JavaScript files are used by Cascade.

- If the parameter is added and is set to N, full versions are used by Cascade.

Web Tailor Tables

Web Tailor uses a number of tables to:

- Control many of the settings in Banner™ Self-Service

- Display menus

- Assign access privileges

This chapter includes the following tables:

Web Tailor Image Information Table (TWGBIMAG)

This table defines images that can be displayed with menu items, Info Text, etc., in Banner Self-Service.

Column Name	Null?	Type	Column Comment
twgbimag_name	Not null	VARCHAR2(120)	Unique name for image
twgbimag_image_url	Not null	VARCHAR2(120)	URL of image (will be the "off" image when part of a javascript-dynamic image)
twgbimag_image_width		NUMBER	Width of image in pixels
twgbimag_image_height		NUMBER	Height of image in pixels
twgbimag_image_on_url		VARCHAR2(120)	URL of image displayed when user's cursor moves over the "off" image
twgbimag_alt		VARCHAR2(120)	Alternate text to be associated with image
twgbimag_status_bar		VARCHAR2(120)	Text to be displayed in window status bar when user's cursor moves over image area
twgbimag_comment		VARCHAR2(255)	Free-form comments about the image
twgbimag_align		VARCHAR2(30)	Alignment of image
twgbimag_border		VARCHAR2(10)	If image is a link hotspot, width of border highlight
twgbimag_vspace		VARCHAR2(10)	Amount of empty space (in pixels) to be displayed above and below the image
twgbimag_hspace		VARCHAR2(10)	Amount of empty space (in pixels) to be displayed to right and left of image
twgbimag_desc		VARCHAR2(120)	Description of image
twgbimag_activity_date	Not null	DATE	Last date modified

LDAP Settings Table (TWGBLDAP)

This table contains the settings LDAP will use to map to Banner Self-Service. This settings are institution-wide.

> **Note:** You must use the new LDAP Administration page in Web Tailor to make any changes to this table because some of the values it contains are encrypted via the Oracle Obfuscation toolkit package.

Column Name	Null?	Type	Column Comment
TWGBLDAP_PROTOCOL	Not null	VARCHAR2(20)	Indicates the current protocol being used for Self-Service authentication. Valid values are: LDAP - LDAP authentication without SSL LDAP_S - LDAP authentication with SSL None - Authentication is provided by the GOBTPAC table in Banner General
TWGBLDAP_SEARCH_IND	Not null	VARCHAR2(1)	Specifies if the LDAP server will perform an anonymous search for the Distinguished Name (DN) for the user. If Y, the search will be performed, if N, the search will not be performed
TWGBLDAP_ACTIVITY_DATE	Not null	DATE	The date that the information in the table was created or last modified
TWGBLDAP_CERTIFICATE_ LOCATION		VARCHAR2(255)	The Secured Socket Layer (SSL) wallet location. Indicates where the SSL is stored
TWGBLDAP_CERTIFICATE_ PASSWD		VARCHAR2(255)	Indicates the location of the SSL certificate password. This field is encrypted via the Oracle Obfuscation utility.

Column Name	Null?	Type	Column Comment
TWGBLDAP_SSL_AUTH_MODE		VARCHAR2(20)	Specifies the SSL Authentication Mode. Valid values are: 1 - No authentication required 2 - One-way authentication required 3 - Two-way authentication required
TWGBLDAP_SERVER_NAME		VARCHAR2(255)	Specifies the location of the LDAP server
TWGBLDAP_PORT		VARCHAR2(20)	Indicates the LDAP port number
TWGBLDAP_SEARCH_BASE		VARCHAR2(255)	Specifies the directory on the LDAP server from which the search should begin
TWGBLDAP_SEARCH_FLTR_SUFX		VARCHAR2(255)	Specifies the filter suffix that should be used in the search or bind
TWGBLDAP_SEARCH_FLTR_PRFX		VARCHAR2(255)	Specifies the filter prefix that should be used in the search or bind
TWGBLDAP_SEARCH_ATTRIBUTE		VARCHAR2(255)	The name of the attribute that contains the Self-Service ID in LDAP

Web Tailor Overrides Table (TWGBOVRR)

This table contains defines the names of stored procedures that you can use instead of some of the delivered code. It provides, in a sense, an alternate API mechanism. For a specific set of standard functions, you can define an alternate set that are called automatically.

This allows Web Tailor to be used with non-Banner applications, such as Executive Self-Service.

Column Name	Null?	Type	Column Comment
twgbovrr_condition	Not null	VARCHAR2(120)	The name of the package.procedure to be replaced. Valid values are: • twbkslib.f_centurypivot • twbkslib.f_dateinputfmt • twbkslib.f_fetchpidm • twbkslib.f_formatname • twbkslib.f_proceedafterlogin • twbkslib.f_userisdead • twbkslib.f_fetchroles
twgbovrr_replace_value	Not null	VARCHAR2(120)	The name of the alternate package.procedure that will be called via dynamic SQL
twgbovrr_activity_date	Not null	DATE	Date last modified

Web Tailor Global User Interface Settings Table (TWGBGLUI)

This table contains settings for the various user interface options across all the Self-Service products.

Column Name	Null?	Type	Column Comment
twgbglui_system_name		VARCHAR2(120)	Name of Web System, displayed in the header bar of each page when in standalone mode
twgbglui_body_bg_color		VARCHAR2(6)	Obsolete, replaced by CSS
twgbglui_body_bg_image		VARCHAR2(120)	Obsolete, replaced by CSS
twgbglui_help_image		VARCHAR2(120)	Global help graphic name
twgbglui_main_menu_name		VARCHAR2(120)	Name of central menu page
twgbglui_header_image		VARCHAR2(120)	Obsolete, replaced by CSS
twgbglui_header_color		VARCHAR2(6)	Obsolete, replaced by CSS
twgbglui__table_header_color		VARCHAR2(6)	Obsolete, replaced by CSS

Column Name	Null ?	Type	Column Comment
twgbglui__table_default_color		VARCHAR2(6)	Obsolete, replaced by CSS
twgbglui_table_label_color		VARCHAR2(6)	Obsolete, replaced by CSS
twgbglui_table_hilite_color		VARCHAR2(6)	Obsolete, replaced by CSS
twgbglui_header_sep_color		VARCHAR2(6)	Obsolete, replaced by CSS
twgbglui_table_dead_color		VARCHAR2(6)	Obsolete, replaced by CSS
twgbglui_font_face		VARCHAR2(255)	Obsolete, replaced by CSS
twgbglui_activity_date		DATE	Date that the record was last modified
twgbglui_submenu_image		VARCHAR2(120)	Graphic to be displayed to the left of full-page menu items that point to a submenu
twgbglui_app_page_image		VARCHAR2(120)	Graphic to be displayed to the left of full-page menu items that point to an application page
twgbglui_css_url		VARCHAR2(255)	URL where the cascading style sheet is stored for that page
twgbglui_help_url		VARCHAR2(255)	URL where the system-level static HTML page is stored
twgbglui_sct_homepage		VARCHAR2(255)	URL where the Powered By logo is stored
twgbglui_error_image		VARCHAR2(120)	Name of the image that identifies system-level error messages
twgbglui_warning_image		VARCHAR2(120)	Name of the image that identifies system-level warning messages
twgbglui_exit_image		VARCHAR2(120)	Obsolete
twgbglui_menu_image		VARCHAR2(120)	Obsolete
twgbglui_back_image		VARCHAR2(120)	Obsolete
twgbglui_required_image		VARCHAR2(120)	Name of the image that identifies the required fields on each page at a system level

Column Name	Null ?	Type	Column Comment
twgbglui_help_css		VARCHAR2(255)	Provides the location of the system-level Help CSS

Web Tailor Web Parameters Table (TWGBPARM)

This table contains a flexible list of parameter/value pairs that control a variety of functions, including:

- Luminis Platform® integration settings

- PIN length for the Self-Service applications

- Credit card payment gateways

- URL for the system-wide CSS

Column Name	Null?	Type	Column Comment
twgbparm_param_name	Not null	VARCHAR2(30)	Name of global parameter used in Web Tailor
twgbparm_param_value	Not null	VARCHAR2(255)	Value of parameter
twgbparm_activity_date	Not null	DATE	Date last modified

Note: Comments may be added for Web Tailor parameters. Comments are set up as infotext as part of `twbkparm.P_DispAllParams`. Please note that the infotext label is limited to 20 characters. Since Web Tailor parameters can be 30 characters in length, the infotext label will only display the first 20 characters of the parameter name.

Web Tailor Login Return Code Base Table (TWGBRETC)

This table contains the name of user-defined login return locations.

Column Name	Null?	Type	Column Comment
twbbretc_return_cde	Not null	VARCHAR2(30)	Login return code

Column Name	Null?	Type	Column Comment
twbbretc_return_location	Not null	VARCHAR2(120)	Location to return to after successful login, either a Web menu or procedure
twbbretc_frameset_ind	Not null	VARCHAR2(1)	Obsolete
twbbretc_menu_ind	Not null	VARCHAR2(1)	Indicates if the return location is a Web menu. Valid values: Y - it is a Web menu N - it is a normal Web procedure.
twbbretc_ovr_startpage		VARCHAR2(120)	Obsolete
twbbretc_desc	Not null	VARCHAR2(120)	Short description of return location
twbbretc_login_forward_ind	Not null	VARCHAR2(1)	Obsolete
twbbretc_activity_date	Not null	DATE	Date last modified

Web Tailor Main Web Page Settings Table (TWGBWMNU)

This table contains the main definitions for all menus and menu items in the Self-Service applications.

Column Name	Null?	Type	Column Comment
twgbwmnu_name	Not null	VARCHAR2(120)	Name of the Web page - should match the name of the package.procedure that generates the HTML
twgbwmnu_desc	Not null	VARCHAR2(120)	Description of the Web page
twgbwmnu_page_title		VARCHAR2(120)	HTML title of the page - appears in browser window title area
twgbwmnu_header		VARCHAR2(120)	Page header, if no title graphic exists
twgbwmnu_header_image		VARCHAR2(120)	Name of main header graphic (e.g. stylized text) for the page
twgbwmnu_l_margin_width		VARCHAR2(10)	Obsolete

Column Name	Null?	Type	Column Comment
twgbwmnu_main_text_width		VARCHAR2(10)	Obsolete
twgbwmnu_r_margin_width		VARCHAR2(10)	Obsolete
twgbwmnu_help_url		VARCHAR2(120)	URL of help page associated with this page
twgbwmnu_bgcolor		VARCHAR2(6)	Obsolete
twgbwmnu_bg_image		VARCHAR2(120)	Obsolete
twgbwmnu_comment		VARCHAR2(255)	Description of page - used to identify page in select boxes
twgbwmnu_back_url		VARCHAR2(120)	URL of Back link for the page
twgbwmnu_back_link		VARCHAR2(120)	Text for the Back link
twgbwmnu_back_image		VARCHAR2(120)	Obsolete
twgbwmnu_back_menu_ind	Not null	VARCHAR2(1)	Indicates if the Back link URL is a database procedure. A value of Y indicates that it is a database procedure, a value of N indicates that it is not
twgbwmnu_module		VARCHAR2(3)	Web system that this page belongs to.
twgbwmnu_enabled_ind	Not null	VARCHAR2(1)	A value of Y indicates that Web access is currently allowed to this procedure or menu. A value of N indicates that it is not.
twgbwmnu_insecure_allowed_ind	Not null	VARCHAR2(1)	A value of Y indicates that users may access this menu or procedure without authentication. A value of N indicates that they cannot.
twgbwmnu_activity_date	Not null	DATE	Date last modified
twgbwmnu_css_url		VARCHAR2(255)	URL where the page-level CSS is stored
twgbwmnu_map_title		VARCHAR2(120)	Title of the map that is created when a menu is generated as a series of links on a page
twgbwmnu_cache_override		VARCHAR2(1)	Page-level setting that controls Web caching
twgbwmnu_exit_image		VARCHAR2(120)	Obsolete
twgbwmnu_menu_image		VARCHAR2(120)	Obsolete

Column Name	Null?	Type	Column Comment
twgbwmnu_source_ind	Not null	VARCHAR2(1)	Indicates if the row is Baseline (B) or Local (L). The default value is B.
twgbwmnu_help_css		VARCHAR2(255)	Indicates where the package.procedure-level Help CSS is stored
twgbwmnu_adm_access_ind	Not null	VARCHAR2(1)	Indicates if the page is secured under VBS or not. Valid values are: Y - Users must enter an Oracle username and password to access the page. When they do, the FGAC and PII rules set up in Banner for each user's role will determine the actions that person can perform. These rules will be in effect for the duration of their sessions. N - Users can access the page without entering an additional ID and password.
twgbwmnu_print_css		VARCHAR2(255)	PRINT STYLE SHEET: The location of page level CSS that controls presentation for printing.

Web Tailor Web Rules Table (TWGBWRUL)

This table contains the settings for rules that will affect the look and feel of all your Self-Service pages and specify how the pages function.

Column Name	Null?	Type	Column Comment
twgbwrul_time_out	Not null	NUMBER	Period of inactivity on the Web which will cause a timeout (in minutes)
twgbwrul_cgibin_dir	Not null	VARCHAR2(60)	Relative URL of the CGI-BIN directory on the Web server
twgbwrul_date_fmt	Not null	VARCHAR2(30)	Default date format for the Web

Column Name	Null?	Type	Column Comment
twgbwrul_time_fmt	Not null	VARCHAR2(30)	Default time format for the Web
twgbwrul_login_attempts	Not null	NUMBER(2,0)	Number of unsuccessful login attempts for the same userid before the account is disabled
twgbwrul_activity_date	Not null	DATE	Date last modified
twgbwrul_error_gif		VARCHAR2(60)	Obsolete
twgbwrul_warning_gif		VARCHAR2(60)	Obsolete
twgbwrul_start_frameset		VARCHAR2(120)	Obsolete
twgbwrul_disp_usage_ind	Not null	VARCHAR2(1)	Indicates if a usage agreement statement is displayed after first login
twgbwrul_exit_gif		VARCHAR2(60)	Obsolete
twgbwrul_pin_exp_days		NUMBER(4,0)	Number of days a user's PIN remains valid
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

Column Name	Null?	Type	Column Comment
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
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 dadb70, and the VBS DAD is dadb70o, this field will contain o.

Web Tailor Web SessionID Table (TWGBWSES)

This table contains session-specific information, including:

• 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

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 publishing process. For example, gi_mybanner is the SourceInfo for the My Banner channel.

Column Name	Null?	Type	Column Comment
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: `Selected` - the Banner Main package, TWBCMAIN, will validated the external system ID `Cleared` - 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: `I` - Informational `N` - 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.

Column Name	Null?	Type	Column Comment
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

Column Name	Null?	Type	Column Comment
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
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

Column Name	Null?	Type	Column Comment
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
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 twtvrole)
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.
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., checkbox, radio 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
twtvmodu_header_caps_on		VARCHAR2(1)	A value of Y indicates that the Header Text in this module is printed with twbkfrmt.P_HeaderCaps. 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 twbkwbis, but not value passed to P_CloseDoc.
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 twgrrole; 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.

TWBKBSSF

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.

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.

`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` - 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 GOBTPAC 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.

`sid`	Student ID
`pin`	Personal Identification string
`newpin`	New PIN number
`verpin`	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.

url   The URL of the new hyperlink

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

`F_PrintAnchor` - Produces `<A>  </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.

`imagename`	The high-level image name, from `TWBKIMAG_NAME`
`cismap`	`ISMAP` 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 `<SPAN>  </SPAN>` 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	class='de<datatype>'
NONTABULAR	class='nt<datatype>'
DATADISPLAY	class='dd<datatype>'
PLAIN	class='pl<datatype>'
MENUPLAIN	class='mp<datatype>'
INFO	class='in<datatype>'
DATAWITHBORDER	class='db<datatype>'

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 htp.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.

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

`imagename`	The high-level image name, from `TWBKIMAG_NAME`
`cismap`	`ISMAP` 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, #)

`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.

`name`	Name of the menu, from `TWBRMENU_NAME`
`display_type`	Current valid values are: `F` - 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. `B` - Body, or full-page, menu. The links are printed in a bulleted format
`font_size`	If the size of the links should be different from the default, send the new size here (e.g., `font_size='-2'` to have the text display two sizes smaller than the surrounding text)
`num_in_row`	By default, for a footer menu, there are 3 links on each row. This is the override for that number.
`validate_links`	By default, the `twbkwbis.f_validlink` 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 `FALSE`, links are printed as is.
`map_title`	Header and footer link menus generated by `p_printmenu` will be enveloped in a MAP tag. The `map_title` parameter will accept a value that will be assigned to the title attribute on the MAP tag.
`ccaption`	Full page menus generated by `p_printmenu` will be enveloped in an HTML table. The ccaption 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).

`message`	The message to be printed.

message_type   Determines how `P_printmessage` 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`	There are two types of boxes used in Web Tailor: `C` - Create and `U` - Update
	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.
`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. Since 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.

`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 - 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_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.

name_in	Name of the menu, from TWBRMENU_NAME
orig_seq_tab	Table of original sequence numbers as printed on the Reorder Menu Items page.
new_seq_tab	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.

name_in	Menu name from TWBRMENU_NAME
return_msg	Return message to be printed.
return_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`	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.
	`N` - prints the Info Text message for this package that has the label `ERROR_NULL`.
	`C` - prints the Info Text message with the label `ERROR_NOCHANGE`.
	`D` - prints the Info Text message with the label `ERROR_DUPLICATE`.
`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`	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.
	Valid values are: `S` - Success, `N` - Error, Record Not Found, `E` - Other Error, `D` - Successful Delete

`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 invalid. 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.

`trans_type`	Transaction type indicator. Valid values: `U` - update, `C` - create.
`*_in`	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 TWBKSERR. 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.

`*_in`	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.

`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_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`	`Y`=`I`=Integer, `C`=Currency, `A`=Alpha, `O`=Alphanumeric, `N`=No check
	Integer checks that all characters are in (0-9), Currency allows for the addition of one period ('.') Alpha allows (a-z, A-Z) Alphanumeric allows (0-9, a-z, A-Z)
`max_length`	`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.
`max_value`	`X` by default - this means that no check is performed. If value is an integer, the value of the field may be no greater than the specified value.

`min_value`	`X` by default - this means that no check is performed.
	If value is an integer, the value of the field may be no less than the specified value.

The other parameters are:

`field_name`	Name of field on the form: `<INPUT NAME="field_name"`...
`field_type`	Type of field to be checked. Valid values are:
	`T` - Text field
	`S` - Select box (A `None`/`NULL` value *must* be first in the `<option>` list for this select box).
`field_desc`	Description of field, to be included in error message if the field is found to be null.

`P_ValidateFormOpen` - Prints an htp.formOpen, with one fixed attribute: a javascript call to a function to be performed at the onSubmit event.

`curl`	URL of the CGI script.
`ctarget`	The frame to load the results of the form submit into.
`cenctype`	The type of encoding used.
`cattributes`	Any additional attributes to be added to the form tag.

`P_ValidateResetButton` - Prints a form reset button. This procedure is used to standardize the text label on validation/data entry forms.

`P_ValidateSubmitButton` - Prints a submit button that validates the data in the form with a JavaScript `onClick` method.

`validate_tab`	PL/SQL table of form fields that need to be checked for not null values.
`num_validate`	Number of fields to be validated.
`button_text`	Will override the default text shown on the button (default is currently **Save Changes**).
`cname`	Name associated with this button.

TWBKWBIS

This package provides the security and menu processes.

> **Note:** All the PIN-related changes to GOBTPAC records are performed
> via 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.

name	The high-level page name, from `TWBKWMNU_NAME`
label	Message label, from `TWBRINFO_LABEL` (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` table.
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 `must` contain parameter PLACEHOLDERS, `<**VALUE1**>` or `<**VALUE2**>`, respectively. These are to be used if user-specific data needs to be displayed as part of a standard error or return message.

`F_FetchWTParam` - Fetches the parameter value associated with the specified parameter name in the TWGBPARM table.

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

str	Character string date variable

`F_formSelectOption` - Function that replaces `htf.formSelectOption`.

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

`F_GenRandId` - Function to generate a random ID.

id	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.

`pidm`	Banner PIDM
`name`	Parameter name
`encr_flag`	*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`.

`check_value`	The string to check (`NULL` or `NOT NULL`)

`F_tableDataOpen` - Function 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

`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`.

`name`	package.procedure

`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`.

`name`	Package.procedure or frameset name
`name_type`	`WMNU` if a Web menu or procedure, `FRAM` if a frameset

`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.

`pidm`	Banner PIDM - this is an OUT parameter. The PIDM retrieved from the session cookie will be passed back here
`insecure_allowed`	If the page calling `F_ValidUser` does not require successful authentication to allow access, this is set to TRUE. Most pages should keep the default parameter of `FALSE`.
`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 `vbsdata` is the VBS DAD.

`P_ChangePin` - Generates the change PIN HTML form.

`msg`	Optional code used to retrieve message strin
`ret_loc`	If `P_ChangePin` 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 `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.

`ret_type`	Will be `M` if the `ret_loc` is a Web menu from TWGBWMNU. `NULL` if the `ret_loc` is not specified, or if it is a procedure (not a menu) from TWGBWMNU.

`P_CloseDoc` - Generates the ending HTML page.

`release_no`	Release number to print at page bottom

The following parameters were added for use by the Student Admissions module:

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

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

`name`	The high-level page name, from `TWBKWMNU_NAME`
`label`	Message label, from `TWBRINFO_LABEL` (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 *must* 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 `htp.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.
`ret_type`	Will be `M` if the `ret_loc` is a Web menu from `twbkwmnu`. `NULL` if the `ret_loc` is not specified, or if it is a procedure (not a menu) from `twbkwmnu`

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

`msg`	Optional code used to retrieve message string
`sid`	User ID

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

msg	Optional code used to retrieve message string
ret_loc	If `P_SecurityQuestion` 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 `TWGBWRUL_BACK_URL` for the `P_SecurityQuestion` procedure. If you use the default, or specify a return location that is a menu, the successful return message will be printed.
ret_type	It will be `M` if the `ret_loc` is a Web menu from TWGBWMNU; it will be `NULL` if the `ret_loc` is not specified, or if it is a procedure (not a menu) from TWGBWMNU.

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

pidm	Banner PIDM
name	Parameter name
val	Parameter value
encr_flag	`Y` 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 TWBKPARM 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. 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:

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" on page 171

- "Web Tailor Menu" on page 172 (an example of a menu page)

- "Customize the selected module" on page 173 (an example of an application page)

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.
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

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.
`pageheaderdiv1`	The DIV class that contains H1, which contains the page header.
`headerlinksdiv`	The DIV class that contains the links to other web pages. It 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

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.
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 `Select another Module to customize`.
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, `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`).

Name	Description
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.