

Penny's Portal Techniques - Navigating the Portal

The benefits of publishing your organisation's information in a corporate portal are not achieved unless the portal navigation mechanism is intuitive and easy to maintain. This article describes a dynamic method for providing portal navigation.

Introduction

There are two main mechanisms used for accessing information in an Oracle Portal site.

- Perform a search
- Use the navigation mechanism provided by the portal

Searching will be considered in a future article.

The two main navigation mechanisms offered as standard by Oracle Portal are as follows:

- Navigate using Tabs
- Drill up and down the page hierarchy

While tabs may work well for an intranet portal with a limited number of pages, they do not provide an easy and intuitive mechanism for external users to access large amounts of information.

The page hierarchy does provide a mechanism for users to drill to levels of detail, however the sub page links available in the product do not provide a particularly attractive interface and have little flexibility.

This article describes the use of a dynamic portlet to create a drop down menu structure which can be used to navigate around the portal.

Designing and Building the Menu

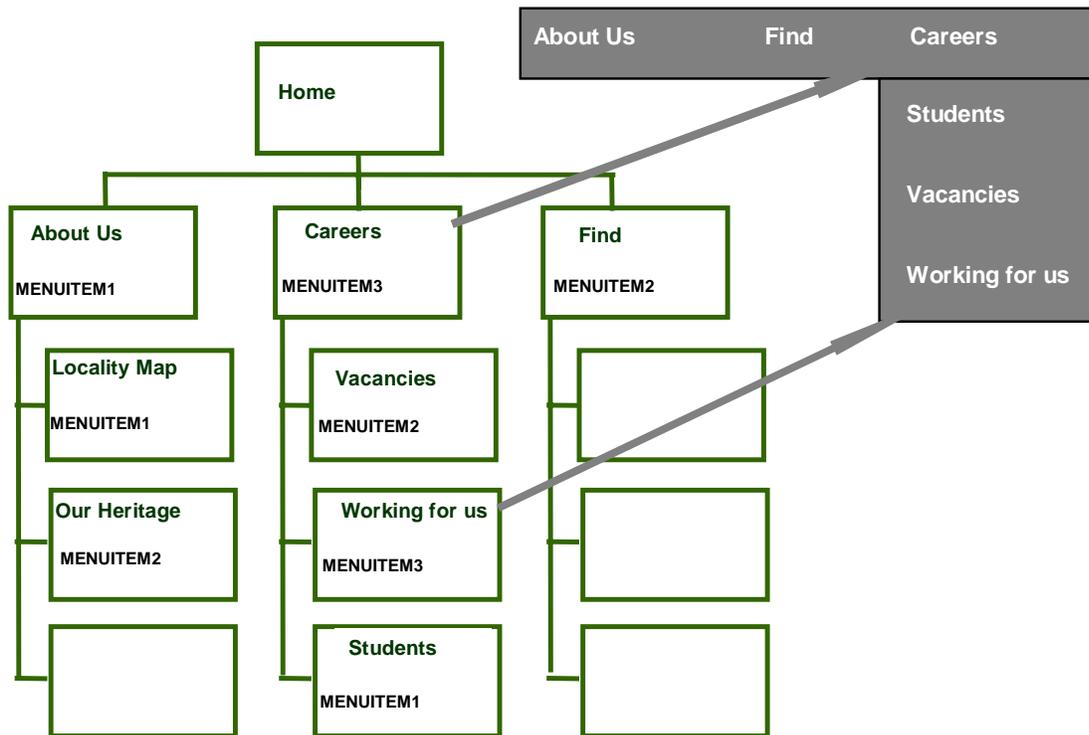
The first step is to decide the hierarchy on which the navigation menu will be based. We have used two approaches successfully at various client sites. The menu can be based on the page hierarchy of a page group, or on a hierarchy of perspectives.

Base the Menu Structure on the Page Hierarchy

Define the Menu Structure

In this approach the menu reflects the hierarchy of pages within the page group. This provides a similar structure to the sub page links provided in the product but allows greater flexibility. In order to define which pages should be included in the menu and the order in which they should be displayed, a number of categories are defined. The categories are named MENUITEM n where n is a number that indicates the position of the page in the menu. Each page that should be included in the menu is assigned one of these categories. *Figure 1* illustrates the use of a page hierarchy and categories to define the required menu hierarchy

Figure 1



Create the Dynamic Menu

The menu is created using Javascript which is generated from the Oracle Portal tables. The components are as follows:

1. A packaged PL/SQL procedure which builds the Javascript code is defined.
2. A dynamic PL/SQL component is used to call the packaged procedure.
3. The PL/SQL component is displayed as a portlet in a navigation page.
4. The navigation page is displayed as a portlet in the standard templates used to build your Oracle Portal site.

I have not included an example of the Javascript code, firstly because it takes up to much space, and secondly because there are as many styles of Javascript menus as there are web developers. The key is to create the menu you want using hard coded Javascript and then use the PL/SQL procedure to create the same code dynamically. We started by selecting the required menu structure into a PL/SQL table using the statement included in *Listing 1*, and then looping through the information stored in the PL/SQL table to create the Javascript code.

Defining Menu Styles

The styles used for the menu can be defined either in a Cascading Style Sheet or as a Portal style. In order to allow the users to easily change the style, using a Portal style is recommended. A specific style is defined in Portal and used only for the menu. The values of the style are selected from the Portal tables in the procedure that creates the menu.

The style components used by the menus should be documented so that the Portal users know which style components to change to effect a change to the menu. *Figure 2* provides an example of this mapping for a horizontal menu bar.

Figure 2 – Example of using standard Portal style components for a menu:

Component	Use
Tab Inactive Color	Menubar back color
Tab Active Color	Menubar active back color
Tab Inactive Text	Menubar font attributes
Tab Active Text	Menubar active font color only
Portlet Heading Color	Horizontal pop back color
Portlet SubHeading Color	Horizontal pop active back color
Portlet Heading Text	Horizontal pop font attributes
Portlet SubHeading Text	Horizontal pop active font color only

Getting the Menu Style

Our approach was to select information from the Oracle Portal style tables into a PL/SQL table. From there the styles can be used in the generation of the menu. *Listing 2* contains the statement requirement to select a specific style from the portal tables. Note that wherever the column SITEID is found in a Portal table this represents the identifier of the Page Group. The ID and NAME of all the page groups can be found in the WWSBR_SITES\$ table.

Maintaining the Menu

Changes to Page Structure

Since the menu is dynamically generated from the page structure, changes to the site can be automatically reflected in the menu without any coding changes. If a page should be removed from the menu it is only necessary to remove the category MENUITEMn from the page. If a new page is added, allocating the MENUITEMn category will automatically include it in the menu.

Changes to Access Privileges

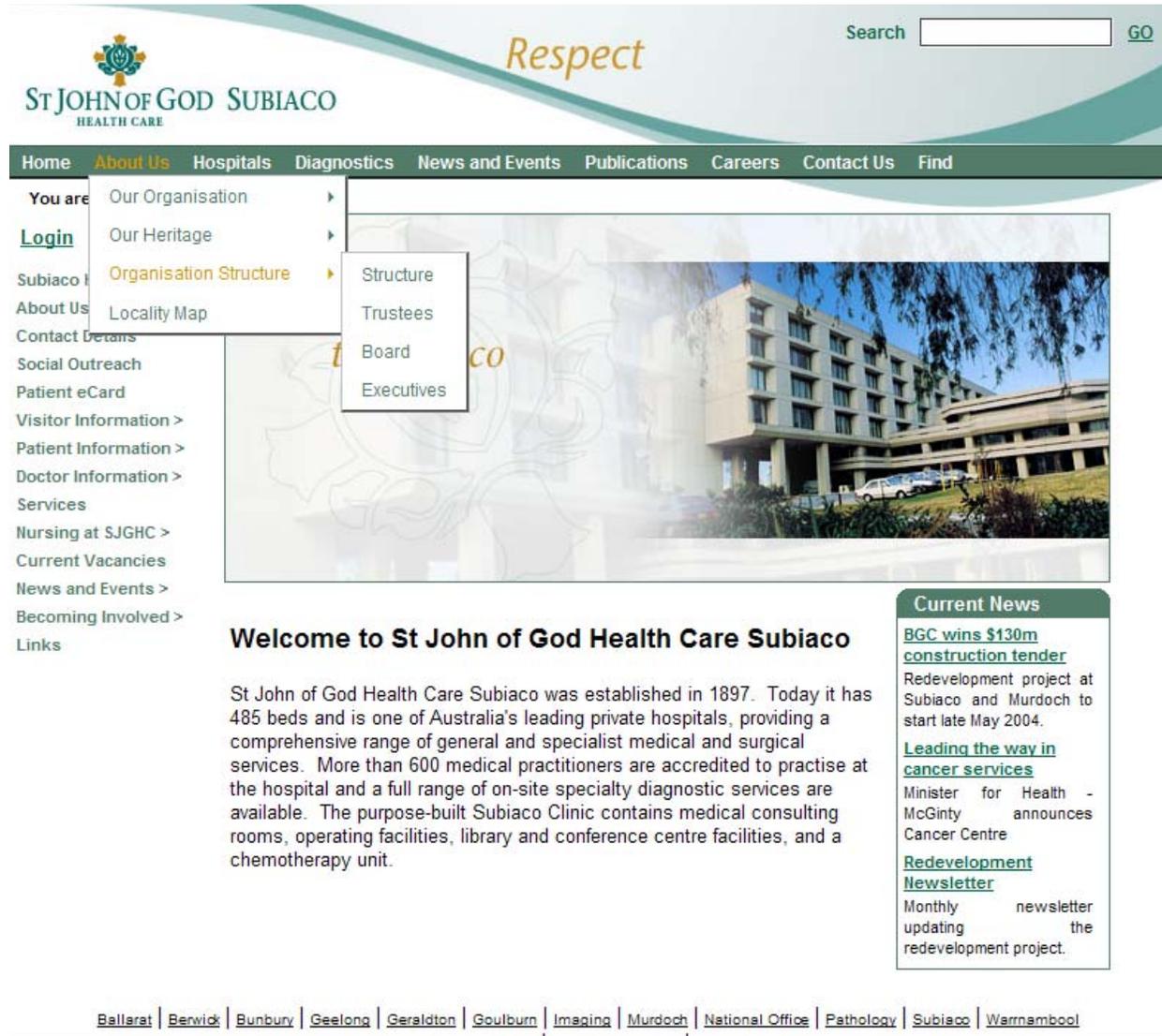
The view WWV_USER_CORNERS is used to generate the menu. This view only displays pages to which the user has access. The menu is therefore responsive to the privileges of the user who is logged on. Any changes to the privileges will be reflected in the menu options displayed.

The Finished Product

Figure 3 is a screen shot of one of Sage Computing Services client's web site in which this technique was used to create an attractive and easily accessed portal site. St John of God Health Care is one of the largest not for profit providers of health care in Australia, and a prominent Catholic health care organisation with operations in Western Australia, Victoria and

New South Wales. Consultants from Sage Computing Services assisted the client's own personnel in creating the portal site.

The dynamic menu technique has been used in two locations on the site. A horizontal menu bar provides navigation for the National pages. The left side menu provides navigation for a specific division of the organisation.



Listing 1 – Select three levels of page hierarchy for given page group

```
CURSOR c_menu (p_page_group wwpob_page$.name%TYPE) IS
SELECT f1.title title1, substr(t1.name,9) num1,
       decode(null,null,'URL/PAGE/'||s.name||'|/'||f1.name) url1,
       f2.title title2, substr(t2.name,9) num2,
       decode(null,null,'URL/PAGE/'||s.name||'|/'||f1.name||'|/'||f2.name) url2,
       f3.title title3, substr(t3.name,9) num3,
```

```

        decode(f3.id,null,null,'URL/PAGE/'||s.name||'/'||f1.name||'/'||
        f2.name||'/'||f3.name) url3
FROM    wwsbr_sites$ s, wwv_user_corners f0, wwv_user_corners f1,
        wwv_user_corners f2, wwv_user_corners f3,
        wwv_topics t1, wwv_topics t2, wwv_topics t3
WHERE   s.id = f1.siteid
AND     s.id = f0.siteid
AND     f0.name = p_page_group
AND     f0.id = f1.parentid
AND     f1.siteid = f2.siteid(+)
AND     f2.siteid = f3.siteid(+)
AND     f1.id = f2.parentid(+)
AND     f2.id = f3.parentid(+)
AND     f1.topicid = t1.id
AND     f2.topicid = t2.id(+)
AND     f3.topicid = t3.id(+)
AND     f1.subscriber_id(+) = f0.subscriber_id
AND     f2.subscriber_id(+) = f1.subscriber_id
AND     f3.subscriber_id(+) = f2.subscriber_id
AND     s.name = p_page_group
AND     t1.name like 'MENUITEM%'
AND     t2.name(+) like 'MENUITEM%'
AND     t3.name(+) like 'MENUITEM%'
ORDER BY to_number(num1), to_number(num2), to_number(num3);

```

Listing 2 - get the individual style values form the Portal style tables

```

FUNCTION style_value (p_name IN wwv_settings.name%TYPE,
                    -- e.g. 'TabBackgroundColor'
                    p_valuetype IN wwv_settings.valuetype%TYPE,
                    -- e.g. 'Background-Color'
                    p_type IN wwv_settings.type%TYPE, -- e.g. 'Tab'
                    p_settingssetid IN wwv_settings.settingssetid%TYPE,
                    p_siteid IN wwv_settings.siteid%TYPE)
RETURN wwv_settings.value%TYPE IS
    v_value wwv_settings.value%TYPE;
    CURSOR c_value IS
    SELECT v_value
    FROM    wwv_settings
    WHERE   name = p_name
    AND     valuetype = p_valuetype
    AND     type = p_type
    AND     siteid = p_siteid
    AND     settingssetid = p_settingssetid;
BEGIN
    OPEN c_value;
    FETCH c_value INTO v_value;
    CLOSE c_value;
    RETURN v_value;
END style_value;

```