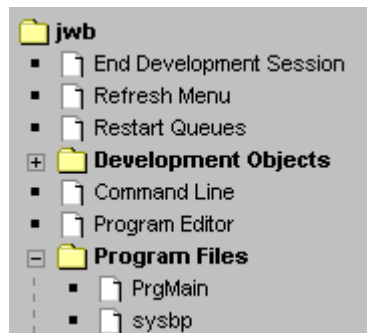


# Creating jWB Menus

## **Introduction**

JBASE for web builders has integrated into it the functionality to build menus that have the same look and feel as those in Window's Explorer. The menu is a visual feature that allows a user to click on a folder icon dropping down the contents of that folder. This is similar to menus in Windows Explorer but jWB also features menu "links" that are not folders. These "links" can be clicked on and have different classes of functionality associated with them.



A snapshot of the JWB menu structure

One way of creating a menu is by hard coding a menu structure with the folders and all the "links" explicitly declared at design time.

The second way is by creating a menu structure from "Creation Subroutines". A creation subroutine is called by jWB when creating a new menu, in the same way a pre-page routine is called when a page is submitted to the server. This means that it can create a menu structure based on user input or a specific files, making it a powerful means to update, change and create dynamic menus on the fly.

Both these methods will be described below with an example of how to build your own.

## **Prerequisites**

This white paper assumes that the user has knowledge of jBASE for Web Builders, knows their way around the development environment, knows how to create a new application module and is familiar with the properties, class and design pane of an object. For creating a menu from a subroutine, you should also be familiar with BASIC code.

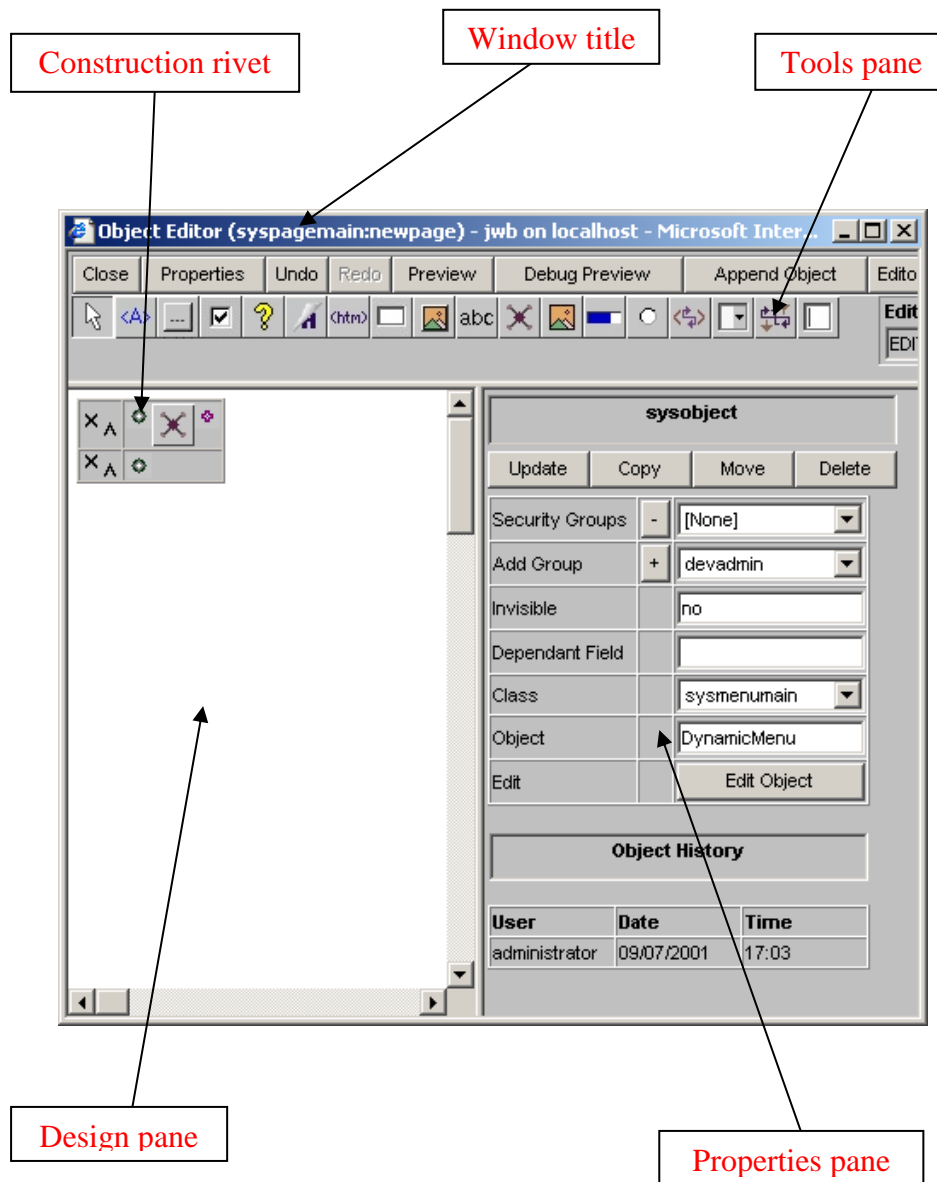
## IMPORTANT

The following documentation shows how to create menu objects and how to get a functioning menu structure. However, this is not very useful unless the menu is on a jWB page. Putting Menu objects on a page requires the following property to be set in the properties pane of that page. (click the properties button at the top of the window)

<b>Header HTML</b>	<code>{{ script  sysstdmenu }}</code>
--------------------	---------------------------------------

The object that you place on the page must also be of the class “sysmenumain” which can be selected from the “class” drop down box in the properties pane.

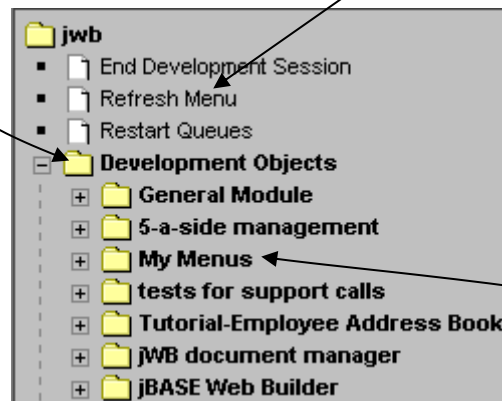
When referencing a jWB Object Editor window the following terms are used in this white paper.



A jWB window with details of terms used in this documentation

## Creating a simple Hard Coded Menu structure

1) After logging into jWB, create a new application module and call it Menus. Give it the description “My Menus”. You must click on the “refresh menus” link to make it show on the Development Objects menu. E.g.



Jwb menu including “My Menus”

2) Click on My Menus and select “Menu Definition”.

3) Add a new object and call it “*HardCodeMenu*”. A window will appear with the title “*sysmenumain:HardCodeMenu*” with various menu **classes** in the **tools pane**. These are described below:





Icon	Class	Description
	Systemnuitm	This is menu call to another menu object
	Systemnujs	This is a link to some Javascript code
	Systemnupge	This is a link to a jWB page
	Systemnuurl	This is a link to a url.

Table of menu classes, their names and their icons

These icons can be selected and placed on the **construction rivet(s)** in the **design pane**

Please refer to this table when building the menu described below.

## Build the Menu Page

4) Select the sysmnuitm icon and place it on the **construction rivet** in the **design pane**

5) Type “AnotherMenu” in the “Menu Name” **properties pane**

6) Click “Edit Menu”.

A new window will appear **titled** “sysmenumain:AnotherMenu”.

7) Select the sysmnujs icon and place the **construction rivet** on the **design pane**.

8) Type the following in the **properties pane**:

<b>Menu Prompt</b>	<i>Sub Menu Alert</i>
<b>Java Script Code</b>	<i>alert('hello again');</i>

9) Click “Update” and then close the window.

10) Select the “sysmenumain:HardCodeMenu” window and place a sysmnujs icon on the **construction rivet** below the “Sub Menu(AnotherMenu)” that you have just been editing.

11) Type the following in the **properties pane**:

<b>Menu Prompt</b>	<i>Alert</i>
<b>Java Script Code</b>	<i>alert('hello');</i>

12) Click “Update”

13) Place a sysmnpge icon on the next **construction rivet** and type in the **properties pane**:

<b>Menu Prompt</b>	<i>goto logon page</i>
<b>Page Name</b>	<i>syslogon</i>

14) Click “Update”

15) Place a sysmnuurl icon on the next **construction rivet** and type in the **properties pane**:

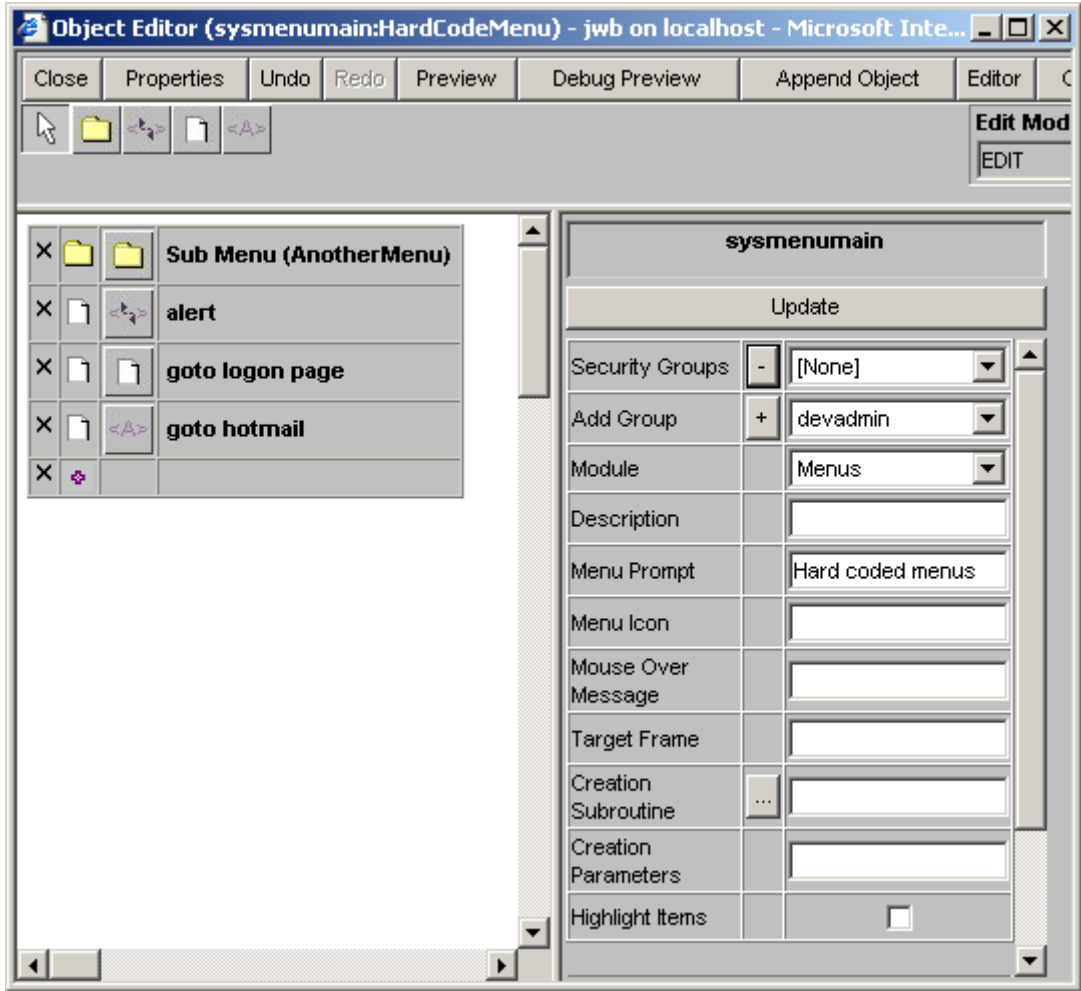
<b>Menu Prompt</b>	<i>goto hotmail</i>
<b>URL</b>	<i>http://www.hotmail.com</i>

16) Click “Update”

17) Finally, click on the “Properties” button above the **tools pane** and type into the **properties pane**:

<b>Menu Prompt</b>	<i>Hard coded menus</i>
--------------------	-------------------------

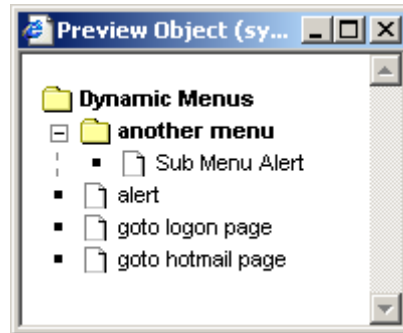
You should now have a window that looks like the following:



How your sysmenumain:HardCodeMenu screen should look

## Preview your menu page

You are now ready to preview your menu screen. Click “Preview” at the top of the window and bar any typing error, you should have a menu screen like below:



Previewing your menu

Moving the mouse over each of the links should cause it to turn blue and become underlined. Clicking on the links should send you to the relevant page, or cause the JavaScript code to execute. If you get any error messages, you may have mistyped some JavaScript or incorrectly spelt various properties in the properties pane of the menu objects.

## Creating a Simple Dynamic Menu Structure

The following example will document building the same menu as the Hard Coded Menu structure, but this time it will be built using two “Creation Subroutines”. (To familiarise yourself with the terms used below, I suggest you follow the instructions for building a Hard Coded Menu). The “Creation Subroutines” are “hard coded” to produce the same results as above, but bear in mind that they could read information from a file or from user input and create menus using these “dynamic” means.

If you haven’t already done so, follow the instructions 1) and 2) in “Creating a Simple Hard Coded Menu structure”.

3) Add a new object and call it “*DynamicMenu*”. A window will appear with the title “sysmenumain:DynamicMenu” .

4) Type into the **properties pane**

<b>Creation Subroutine</b>	<i>BuildDynamicMenus</i>
----------------------------	--------------------------

And hit the  button next to it

A new editing widow will appear with the title “(PrgMain,BuildDynamicMenus)”. You can copy and past the subroutine titled “builddynamicmenus” below, but it is based on the following skeleton code:

```
*--- parameter (menurec) is not html as per usual but an item
*--- that will contain a menu record
    SUB skeletoncode(menurec)
*--- include the common block
    INCLUDE sysbp syscommon
*--- prepare objects for amend. Compusary code
    mobj = RAISE(menurec<12>)
*--- loop and process
    FOR X = 1 TO Y
*-----This is where you can loop and read details from a file
*-----and then build up a menu based on these results
        mrec = "Any valid sysmnutype"
*----- these items must be defined as per their object definitions
        mrec<10,1> = "menu prompt"
        mrec<10,4> = "menu parameter"
*----- these lines are compulsory to store the object
*----- lower it for storage
        mrec = LOWER(LOWER(LOWER(mrec)))
        mobj<-1> = mrec
*--- get next set of data
    NEXT X
*--- this line is compulsory
*--- restore objects
    menurec<12> = LOWER(mobj)
*--- done
    RETURN
```

Systemtype doesn’t actually exist, but can be sysmnitm, sysmnujs, sysmnpuge or sysmnuurl

Details of these parameters can be found in the developer reference pages 72 - 75

## BuildDynamicMenus subroutine

```
SUB builddynamicmenus (menurec)
*--parameter is not html as per usual but an item that will contain a menu record
*--this subroutine will build a menu structure like the Hard Coded Menu structure

*--Include the common block
    INCLUDE sysbp syscommon

*--Initialise some variables
    mrec = ""; mobj = ""

*--this line compulsory to prepare menu objects for amend
    mobj = RAISE (menurec<12>)

*--The first object is a sysmnuitm that will hold the "Sub Menu Alert". The "LOWER..."
*--line is compulsory. We store the results in mobj, and then assign this to menurec.
    mrec = "sysmnuitm"
    mrec<10,1> = "AnotherDynamicMenu"
    mrec = LOWER (LOWER (LOWER (mrec)))
    mobj<-1> = mrec

*--the following produces the alert menu
    mrec = "sysmnujs"
    mrec<10,1> = "alert"
    mrec<10,4> = "alert('hello');"
    mrec = LOWER (LOWER (LOWER (mrec)))
    mobj<-1> = mrec

*--the following produces the 'goto logon' menu
    mrec = "sysmnujpe"
    mrec<10,1> = "goto logon page"
    mrec<10,4> = "syslogon"
    mrec = LOWER (LOWER (LOWER (mrec)))
    mobj<-1> = mrec

*--the following produces the 'goto hotmail' menu
    mrec = "sysmnuurl"
    mrec<10,1> = "goto hotmail page"
    mrec<10,4> = "http://hotmail.com"
    mrec = LOWER (LOWER (LOWER (mrec)))
    mobj<-1> = mrec

*--this line is compulsory
*--restore objects
    menurec<12> = LOWER (mobj)
    menurec<10,1> = "Dynamic Menu"
*--done
    RETURN
```

These four sections of code are the equivalent of the four menu objects that we placed on the design pane in the section Hard Coded Menu Structure

This names the menu

5) When you have copied and pasted the code above, click on the button “Update and Compile” and make sure that the code compiles correctly.

You may have noticed that in the code above, there is a reference to “AnotherDynamicMenu”. This code defines another menu item that we must now define.

6) Create another menu object under “My Menus” and call it “AnotherDynamicMenu”. A window will appear with the **title** “sysmenumain:DynamicMenu” .



7) Type into the **properties pane**

<b>Creation Subroutine</b>	<i>BuildAnotherDM</i>
----------------------------	-----------------------

And hit the  button next to it

8) In the editing window, paste in the following code:

```
SUB buildanotherdm(menurec)
*--instead of html, the subroutine variable is a record
*--This produces "another menu" with the "Sub Menu Alert"

*--Include the common block
INCLUDE sysbp syscommon

*--Initialise some variables
mrec = ""

*--Compulsary statement
mrec = RAISE(menurec<12>)

*--assign it as a javascript object
mrec = "sysmnujs"

*--assign the name and javascript to the menu object
mrec<10,1> = "Sub Menu Alert"
mrec<10,4> = "alert('Hello again');"

*--Compulsary code
mrec = LOWER(LOWER(LOWER(mrec)))

*--Give the "folder" object a name
menurec<10,1> = "another menu"

*--Compulsary code
menurec<12> = LOWER(mrec)

RETURN
```

This produces the second JavaScript alert in the sub menu

We name the sub menu here

9) Click on “Update and Compile” and make sure the code compiles OK.

If you now preview “DynamicMenu” you should get the same menu structure and functionality as the Hard Coded menu.