

jWB – Creating A Development Class

INTRODUCTION

jBASE for Web Builders comes complete with a feature rich set of development classes, which should prove sufficient for the majority of users to successfully develop transactional web sites. However, it is possible, and often desirable for users to develop their own classes.

This functionality allows users to tailor the IDE to their own requirements as well as providing the flexibility to allow jWB keep up to date with the changing requirements of the web development market.

This document explains how a user can develop these new classes and integrate them with the jBASE for Web Builders development environment.

OVERVIEW

jBASE for Web Builders development classes are made up of a development routine, typically prefixed 'syse', and a run time routine, typically prefixed 'syso', as well as Selected and Unselected icons. These routines and icons specify what is displayed during editing, when a user selects a development class and drops it onto a page, and at run time, when the page is rendered.

Each class also has a defined set of properties, which are displayed in the properties pane when an object of that particular type is selected.

BUILDING THE DEVELOPMENT CLASS

The web builder class is going to allow us to include a '*Mail To*' anchor tag, which will allow the user to send an email to a specified address.

Start up jWB, then expand the '*Configuration*' menu and click on '*Development Classes*'. This will display a list of the existing development classes. Click on '*Add a New Item*' then enter '*usrmmailto*' when prompted. This will open the '*Editing Class*' screen for the new class.

In the description field enter '*MAILTO anchor Tag*'. In the '*Run Time Routine*' field enter '*usromailto*' then click on the button next to the field to open an edit window.

This routine will be called when the page is rendered and generates the html for the '*Mail To*' objects.

The html starts with '*<a*' which indicates that this is an anchor tag. The '*href*' for the target email address is then generated by concatenating the text '*mailto:*', indicating a mail-to tag, with the '*Email Address*' property of the object.

The routine then checks if any of the other property fields have been filled in and adds them to the html tag where appropriate. Finally, the routine inserts the contents of the '*Text*' property and closes the tag with '*/a>*'.

Paste in the following routine:

```
SUB usromailto(html)
*--- include the common block
INCLUDE sysbp syscommon
*--- start the anchor tag
html = \<a\
*--- set the href if exists
IF sysobjrec<10,2> = "" THEN
    html := \ href=mailto:"javascript:void(0)"\
END ELSE
    html := \ href='mailto:\:sysobjrec<10,2>:\'\
END
*--- style
IF sysobjrec<10,3> THEN
    html := \ style="\:sysobjrec<10,3>:\"\
END
*--- class
IF sysobjrec<10,4> THEN
    html := \ class="\:sysobjrec<10,4>:\"\
END
*--- insert focus event
IF sysobjrec<10,5> # "" THEN
    html := \ onfocus="\:sysobjrec<10,5>:\"\
END
*--- insert change event
IF sysobjrec<10,6> # "" THEN
    html := \ onchange="\:sysobjrec<10,6>:\"\
END
*--- insert blur event
IF sysobjrec<10,7> # "" THEN
```

```

        html := \ onblur="\:sysobjrec<10,7>:\\"
    END
*--- add the text (taking language into account
    captiontext = ""
    IF sysobjrec<10,1> # "" THEN
*----- set default position for field statement in case where no
multi-language
        pos=1
*----- check if this user has a non-default language specified
        IF sysuserrec<23> # "" THEN
*----- check if the language id is stored numerically or as
text
            IF NUM(sysuserrec<23>) THEN
                pos = sysuserrec<23>
            END ELSE
                LOCATE(sysuserrec<23>,sysparms,22;pos) ELSE pos = 0
            END
*----- increment location due to first item being default
english text
        pos+=1
        END
        captiontext = FIELD(sysobjrec<10,1>,CHAR(129),pos)
*----- if text is blank then default to english text
        IF captiontext = "" THEN
            captiontext = FIELD(sysobjrec<10,1>,CHAR(129),1)
        END
    END
    html := \>\:captiontext:\</a>\
*--- thats it
    RETURN

```

Update and Compile the routine.

In the '*Unselected Icon*' and '*Selected Icon*' fields, enter a path to the icons you intend to use. These are the icons, which will appear in the jWB toolbar at the top of the screen.

Next enter '*usremailto*' into the '*Edit Routine*' field, then click on the button next to the field to open an edit window.

This routine will determine what happens during the edit process i.e. when the user adds an object of this type to the page or clicks on an existing '*Mail To*' object.

What will be displayed on the screen will either be the contents of the '*Text*' property, if a value has been entered, or the default '*Selected Icon*'.

It also adds an *'onclick'* event to the object so that when a user clicks on the particular object, the relevant properties are displayed in the *'Properties Pane'*.

Paste in the following routine:

```

        SUB usremailto(html)
*--- include the common block
        INCLUDE sysbp syscommon
*--- open the files
        CALL sysopen("sysclasses",sysclasses,errhtml)
*--- generate the html
        html = ""
        alttxt = sysobjrec<10,1>
*--- if the text is blank
        IF sysobjrec<10,1> = "" THEN
*----- get the default image from the classes file
        READ clsrec FROM sysclasses,sysobjrec<1> THEN
            img = clsrec<3,1>
            IF INDEX(img,"{",1) > 0 THEN
                CALL sysftn(img)
            END
            html := \:\','EDIT')"\
        html := \ alt="\:alttxt:\" style="cursor:hand"\
        html := \ border=0 align=top>\
        END ELSE
*----- move to properties on click
        leadin = \<span onclick="ObjLaunch('usrmailto','\
        leadin := syshidden<1>:\','EDIT')" style="cursor:hand"\
*----- add title
        titletext=""
        leadin := \ title="\:titletext:\">\
        html = leadin:\<font
color=#0000FF><u>\:FIELD(sysobjrec<10,1>,CHAR(129),1)[1,50]:\</u>
</font></span>\
        END
*--- return with html
        RETURN

```

Update and Compile the routine.

The four check boxes determine where the component will appear within jWB. The page, menu and table sub class tick boxes reflect whether the class will appear in the toolbox when editing a page,

menu or table. The stand-alone check box indicates that the class will appear in the main jWB development menu. Check the '*Stand Alone*' checkbox, and then enter the following properties.

Description	Type	Pos
Security Groups	Security Field	
Module	Module Field	
Description	Text Box	2
Text	Language	10.1
E-Mail Address	Text Box	10.2
Style	Text Box	10.3
Style Class	Text Box	10.4
Focus Event	Text Area	10.5
Click Event	Text Area	10.6
Blur Event	Text Area	10.7

Finally click on the '*Update Class*' button and the development class should now be created.

Document Created 19th July 2001
Author: Simon Rees