Creating a Shop with

NetObjects Fusion 3.0 and Shop@ssistant

By The Floyd Consultancy Ltd

This manual was created by

The Floyd Consultancy Ltd
Pointers, Wildmoor Lane
Sherfield-on-Loddon
Hook, Hampshire
RG27 0HA
UK

Email: info@floyd.co.uk
Web: www.floyd.co.uk

Phone: +44 (0) 1256 880770
Fax: +44 (0) 1256 880330

Contents

Overview

Overview

The aim of this guide is to show you how to create a Shop using Shop@ssistant and NetObjects Fusion 3.0.

To help you create your site a number of special components have been created by Ingo Fischer (email: ingo.fischer@mata.fzk.de).These Fusion components are also compatible with NetObjects Fusion 2.0.

In the 'Advanced Functions' section of this guide we look at some examples of more sophisticated pages using some additional scripts, but these examples are only compatible with Fusion 3.0, due its more flexible approach to Forms.

Shop@ssistant

Shop@ssistant is the leading system for businesses looking to sell products on the net, and is particularly designed to meet the needs of SMEs.

The system is designed to be simple to integrate into almost any web site design, yet provides one of the most flexible systems available for the Internet merchant.

NetObjects Fusion 3.0

In the crowded world of HTML development tools, NetObjects Fusion stands out as one of the most comprehensive site development tools available.

The Shop@ssistant demo site was created using Fusion, as is our on-line shop for purchases of the product, found at http://www.floyd.co.uk

Getting Started

Fusion Components

The Shop@ssistant Components

The components that are described in this documentation have been developed by Ingo Fischer (email: ingo.fischer@mata.fzk.de) for use with the Shop@ssistant e-commerce system. The components are provided subject to the terms of the Apollon-Components Licence Agreement.

They can be downloaded from The Apollon-Components website at http://www.apollon.de. They are also available from the Floyd Consultancy website http://www.floyd.co.uk and are included the utilities/fusion directory on the Shop@ssistant distribution CD.

Installing the Components

Depending on where you got these components from, you will either have downloaded them from the web, or have installed them from the Shop@ssistant CD; either way you will have up to four installers (one for each component).

To install these components just launch each installer and follow the onscreen instructions.

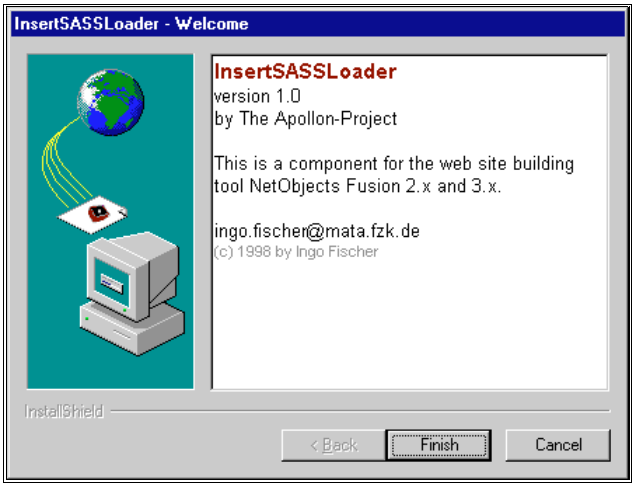

Figure 1

On your START menu you will find a new entry for Apollon-Components, including the latest information and an uninstaller if required.

Apollon-Components License Agreement:

This software is FREEWARE.

Apollon-Components License and Limited Warranty

This legal document is an agreement between you, THE LICENSEE, and Ingo Fischer, THE LICENSOR.

By executing the program files you agree to be bound by the terms of this agreement, which includes the Software License, Limited Warranty and Acknowledgement. If you do not agree to the terms presented in this agreement, DO NOT OPEN the program files.

SOFTWARE LICENSE

Limitations on Reverse Engineering, De-compilation, and Disassembly. You may not reverse engineer, de-compile, or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

Software Transfer. You may permanently transfer all your rights under this License, provided that the recipient agrees to the terms of this License.

Miscellaneous: This License shall be governed and construed in accordance with the laws of Germany.

LIMITED WARRANTY AND DISCLAIMER OF WARRANTY

The software and accompanying written material (including instructions for use and this document) are provided "as is" without warranty of any kind. Further, THE LICENSOR does not warrant, guarantee or make any representations regarding the use, or the results of use, of the Software or written materials in terms of correctness, reliability, accuracy or fitness for purpose.

The entire risk as to the results and performance of the software is assumed by you.

Neither THE LICENSOR nor anyone else who has been involved in the creation, production or delivery of this product shall be liable for any direct, indirect, consequential or incidental damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising out of the use or inability to use this product even if THE LICENSOR has been advised of the possibility of such damages.

This does not affect your statutory rights.

This Limited Warranty shall be governed and construed in accordance with the laws of Germany.

ACKNOWLEDGMENT

By opening the program files you acknowledge that you have read this License and Limited Warranty, understand them and agree to be bound by their terms and conditions.

Shop@ssistant

Working with Shop@ssistant

If you are using Fusion to create your Shop@ssistant shop we recommend that you set your site to Publish directly to the 'pages' directory on your local copy of Shop@ssistant.

Note: Do NOT use Fusion to edit the Shop@ssistant system pages (those pages in the system directory). These files may only be edited with a text editor such as TextPad (provided on the SASS distribution CD) or WordPad (normally found in the Accessories section of the Programs area of the Start menu).

Working with Components

A basic product page

There are four basic elements that make a page on your web site sell your products.

1) Product information – Before your potential customers can purchase they need to be made comfortable with the product. This can only be done by providing them with the information they need to make the purchasing decision (so don't skimp on the product information and images). Fusion can make laying out your information simple.

2) An 'Add to Basket' button – Without this your customers cannot purchase your product. This is the link from your product to the shopping basket.

3) A 'Review Basket' button – It should not be necessary for a shopper to add a product to his basket in order to get to the Review page and thence to the Cashier section. It is strongly recommended that you provide a 'Review Basket' button on each shopping page of your site, as a minimum. Note:- you could add this button to your Fusion MasterBorder design.

4) The Terms & Conditions button – This is an optional addition to your page, but is highly recommended as it provides a simple method and an encouragement to your customers to view and agree the terms. Note:- You could also add this button to your MasterBorder design.

Adding Components to your pages

All the Shop@ssistant components are accessed in NetObjects Fusion *via* the standard 'Component Tools' toolbar which can be selected through the View/Toolbars/Component Tools menu.

The toolbar should appear as below.

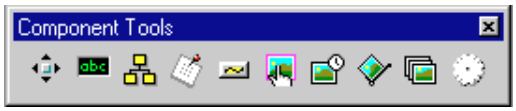

Figure 2

Click on  to insert a Fusion component.

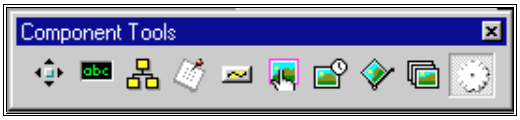

Figure 3

With this option selected the mouse pointer will change to a cross hair when over your page layout.

Now place the mouse where you would like to place your Fusion *Shop*@ssistant Component and click and drag a small bounding box.

When you release the mouse, Fusion will display a dialog box similar to the one below, from which you may select the component you wish to insert.

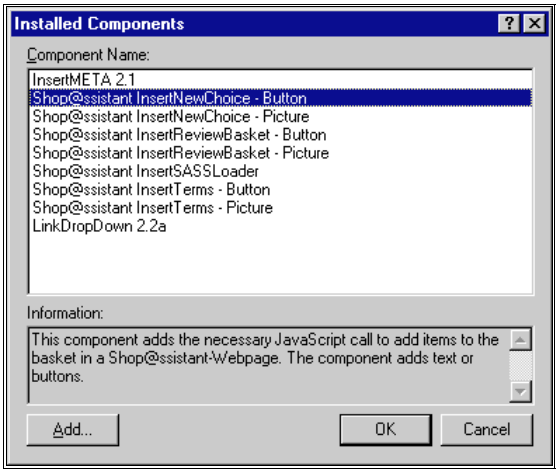

Figure 4

Important Note: These components are only fully functional when the site is published. When viewed using the 'Site Preview' option in Fusion these components will only display as a grey box.

The InsertNewChoice Component

The InsertNewChoice component is used to create an 'Add to Basket' button on your product pages. There are two components that can add this function.

Shop@ssistant InsertNewChoice – Button

Using this component inserts a buying link from a forms button or from a hyperlink.

Shop@ssistant InsertNewChoice - Picture

Using this component creates a buy button using a picture as the hyperlink.

In the following example we are going to guide you through installing a simple forms-style Buy Button. Highlight this option and click on the OK button.

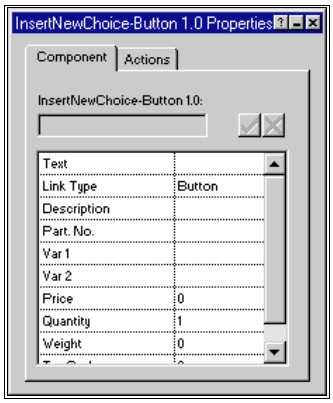

Figure 5

You are now presented with a dialog box with a number of options for you to enter.

Option	Description
Text	This is the text that will appear on the button or as the hyperlink.
Link Type	This enables you to select to use a forms button or a text link.
Description	The description of the product (will be displayed in the basket).
Part. No.	Optional Part No. field (will be displayed in the basket).
Var 1	Optional descriptive variable e.g. colour, size, version (will be displayed in the basket).
Var 2	Second optional descriptive variable e.g. colour, size, version (will be displayed in the basket).
Price	Cost per unit.
Quantity	Quantity being ordered, by default = 1
Weight	Used to calculate shipping costs, normally in kg.
Tax Code	This is dependent on the tax codes and tax rates that have been configured (defaults 0 = Zero Rated while 1 = standard rate tax).

The Link Type has two possible selections: ‘Button’ or ‘Text Link’, either of which may be selected from the drop-down menu.

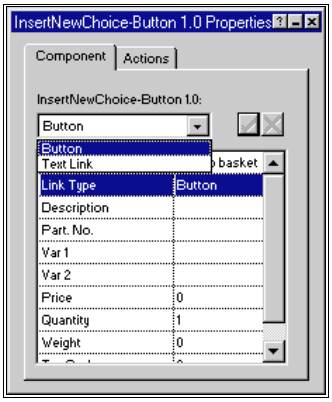

Figure 6

Either of these will display in Fusion as a button, Figure 7.

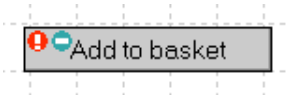

Figure 7

but when published, if ‘Text Link’ was selected, a standard hyperlink will be produced instead, Figure 8.

Add to basket

Figure 8

Figure 9 is an example of the input dialog with all fields completed.

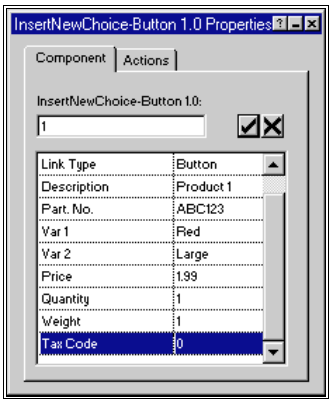

Figure 9

Once you have completed the input of your product options in the dialog box you may click anywhere on the page to continue. To return to this dialog just click on the button/hyperlink and the ‘Layout Properties’ dialog will display the InsertNewChoice options.

The Code Generated

The example above would generate the following code in the page for the Shop@ssistant function:-

```
<FORM><INPUT TYPE="BUTTON" VALUE="Add to basket"
onClick="javascript:top.newchoice('Product
1','ABC123','Red','Large',1.99,1,1,0)"></FORM>
```

Note: some of the comments have been removed to aid understanding and the code line reproduced here is 'wrapped ' to fit the page size.

If the 'Text Link' option had been selected it would have generated the code as follows:-

```
<A HREF="javascript:top.newchoice('Product
1','ABC123','Red','Large',1.99,1,1,0)">Add to
basket</A>
```

Dummy Function

When this component is added to the page, dummy functions are also created between the <HEAD> tags in the document. These are there as 'just in case' functions so that any shoppers who enter the shop without loading the basket will not generate error messages when trying to order. (see the Shop@ssistant installation manual for more information).

```
<SCRIPT>
<!--
function newchoice() {}
function reviewbasket() {}
function terms() {}
function qty_fix() {}
function qlock() {}
function noremove() {}
function parser() {}
function listtext() {}
function listvalue() {}
// Shop@ssistant Dummy functions by Ingo Fischer
(http://www.apollon.de) -->
</SCRIPT>
```

These functions are only added by the first component used on each page.

The InsertReviewBasket Component

The InsertReviewBasket component is used to create 'Review Basket' buttons on your product pages. There are two styles of button available created by two versions of this component:-

Shop@ssistant InsertReviewBasket - Picture

Using this component creates a button using a picture as the hyperlink.

To create a 'Review Basket' button with an image as the button, select the Picture version of the component, and click on the OK button. (Figure 10).

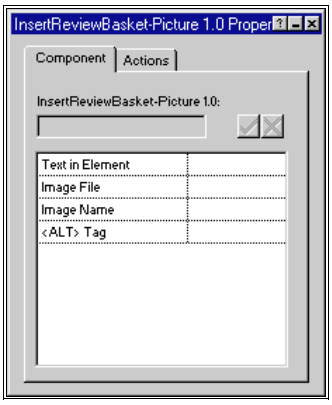

Figure 10

Option	Description
Text in Element	Any text you put in here will be overlaid on your image in the default type style in black. In most cases you would leave this blank and use a custom-designed button if you wanted full control over the appearance of your chosen text.
Images File	This is name of the image file that you are going to use for this button. Click on the ... to locate this file.
Image Name	This is an optional field used to give the image a name so that it could be used as a target for other functions.
<ALT> Tag	This is the text that will be put in the <ALT> tag for this image.

The Code Generated

The example above would generate the following Shop@ssistant function code.

```
<A HREF="javascript:top.reviewbasket()"><IMG
id="InsertReviewBasketPicture1.02" HEIGHT=35 WIDTH=155
SRC="./a_whitebutton.gif"  BORDER=0  ALT="Click here to Review your
basket"  name=Review></a>
```

Shop@ssistant InsertReviewBasket – Button

Using this component creates a button with a browser's forms-style button or creates a hyperlink to take the shopper to the basket review.

To create a 'Review Basket' button select the Button version of the component from the drop-down, and click on the OK button.

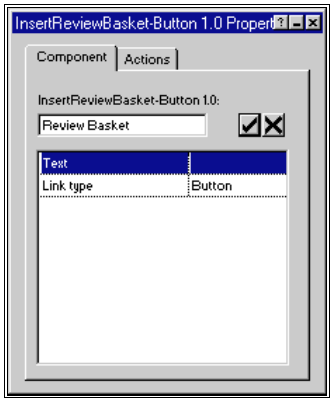

Figure 11

In the InsertReviewBasket properties dialog box there are two options.

Option	Description
Text	This is the text that will be displayed on the Form Button or as the text for the hypertext link.
Link type	This enables you to select the type of button/link you are going to use.

Both will display in Fusion as a button , Figure 12.

Figure 12

but when published, if 'Text Link' was selected, then a standard hyperlink would be produced.

The Code Generated

The example above would generate the following Shop@ssistant function code.

```
<FORM><INPUT TYPE="BUTTON" VALUE="Review Basket"
onClick="javascript:top.reviewbasket()"></FORM>
```

Note: some of the comments have been removed to aid understanding.

If the 'Text Link' option had been selected it would have generated the code as follows:-

```
<A HREF="javascript:top.reviewbasket()">Review Basket</A>
```

The InsertTerms Component

The InsertTerms component is used to create 'Terms & Conditions' buttons on your product pages. Again there are two versions of the component that can add this function.

Shop@ssistant InsertTerms – Button

Using this component creates a button with a browser's forms-style button or creates a hyperlink.

Shop@ssistant InsertTerms - Picture

Using this component creates a button using a picture as the button image.

In this case we are going to select the Button version of the component, so highlight this option and then click on the OK button.

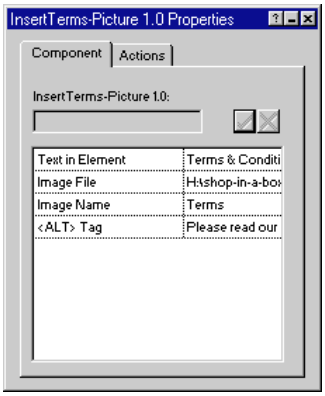

Figure 13

As we are going to use an image for our link to the 'Terms & Conditions' page in Shop@ssistant we have a few extra options in the Properties dialog box to complete.

Option	Description
Text in Element	Any text you put in here will be overlaid on your image in the default type style, in black. In most cases you would use a custom-designed button if full control over the text appearance was important.
Images File	This is the image file that will be used for this button. Click on the  to locate this file.
Image Name	This is an optional field used to give the image a name so it can be used as a target for other functions.
<ALT> Tag	This is the text that will be put in the <ALT> tag for this image.

With the options selected above, the following button is created.

Figure 14

The Code Generated

```
<A HREF="javascript:top.terms()"><IMG id="InsertTermsPicture1.02"
HEIGHT=35 WIDTH=155 SRC="./a_whitebutton.gif"  BORDER=0
ALT="Please read our terms &amp;amp; conditions"  name=Terms></a> <!--
Shop@ssistant InsertTerms 1.0 by Ingo Fischer
(http://www.apollon.de) -->
```

The InsertSASSLoader Component

The InsertSASSLoader component is used to add a special code between the <HEAD> and </HEAD> tags on your page.

The inserted code is designed to test if the browser can run the Shop@ssistant software (i.e. if it is a browser version 3 and above), then load the Shop@ssistant system if it is not already loaded and finally return the shopper to the page they had selected.

This enables your site to be searched by the Search engines like AltaVista and HotBot etc. and have all your pages indexed, improving your visibility to your marketplace. When all of your site pages are submitted to the search engines, or when a customer saves a 'bookmark', there is the possibility that a shopper will re-enter the site on a subsequent visit at a place other than the index page. By incorporating the scripts generated by this Fusion component, when a shopper links directly to a page in your site without loading the home page (and therefore Shop@ssistant), the code detects this and automatically loads the basket.

Note: when developing your site with this component, remember that your pages will automatically try to load *Shop*@ssistant when you want to test them as 'stand-alone' pages. You may find it more convenient to add this function when your site is complete or to leave the code switched off (Figure 15) whilst you are simply checking your HTML pages.

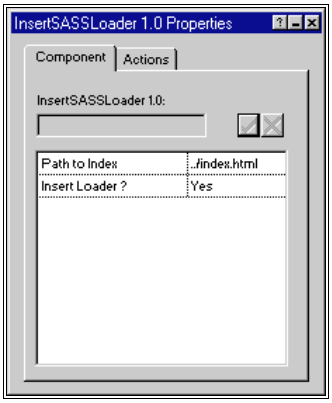

Figure 15

When you insert this component, you will get the above properties dialog box displayed with two input options.

Option	Description
Path to Index	This is the relative path from the current page in the final completed site to the Shop@ssistant index page. This could vary depending on the selected 'Directory Structure' used when publishing your site and for each level in your directory hierarchy (if more than one). The default ../index.html would be fine for a site published using the 'Flat' option when published. For the next level down in the hierarchy, the equivalent code would be: ../../index.html
Insert Loader ?	This is an option switch to enable you to publish your site without the loader code for local testing.

The component will be displayed on the Fusion page layout as a small grey square.

Figure 16

Note: In some sites it would be possible, if using the 'flat' directory structure, to put this in the MasterBorder.

The Code Generated

The following code is an example of what is inserted between the <HEAD> tags when the site is published.

```
<SCRIPT>
<!--
var nappN=navigator.appName;var nappV=navigator.appVersion;var
NN=(nappN=="Netscape"&&parseInt(nappV.substring(0,1))>=3);var
IE=(nappN=="Microsoft Internet
Explorer"&&parseInt(nappV.substring(0,1))>=2);var jsok=(NN||IE);

if (jsok&&(!(window.name==’shopping’ ||
parent.window.name==’shopping’))) {


document.cookie="LOADTHISPAGENOW="+escape(window.location.href)+";
path=/";

  top.location.href="../index.html";

}

// Shop@ssistant InsertSASSLoader 1.0 by Ingo Fischer
(http://www.apollon.de) -->

</SCRIPT>
```

Other Methods

Using Image Maps

It is simple with Fusion to use an image map as the order button. In Figure 17, we have three products each of which has been mapped so that by clicking on the product you will add that product to the basket.

Figure 17

Once you have your image inserted in your Fusion page, create your first image map using Fusion's hot-spot tool. (See the Fusion manual for the basics of creating image maps).

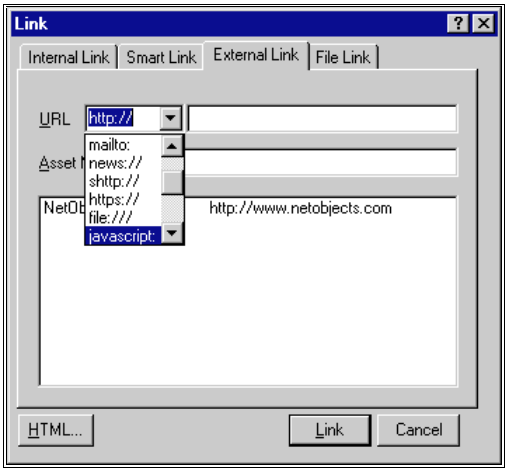

Figure 18

As you complete each image map/hotspot the link dialog will appear. What you need to create is a JavaScript link to the basket for the selected product.

First select the External Link tab and then, in the URL selection, choose JavaScript. Now simply enter the product information.

i.e. top.newchoice ('Description', 'Part. No.','Var 1','Var 2',Price,Quantity,Weight,Tax Code)

Option	Description
Description	The description of the product (will be displayed in the basket).
Part. No.	Optional Part No. field (will be displayed in the basket).
Var 1	Optional descriptive variable e.g. colour, size, version (will be displayed in the basket).
Var 2	Second optional descriptive variable e.g. colour, size, version (will be displayed in the basket).
Price	Cost per unit.
Quantity	Quantity being ordered; by default =1
Weight	Used to calculate shipping costs, normally in kg.
Tax Code	This is dependent on the tax codes and tax rates that have been configured in *Shop*@ssistant.

So in our example we could enter:-

top.newchoice ('Coconut Milk', 'NUTS001','','',1.99,1,1,1)

Note: Variables 1 and 2 are blank in this example, but could contain any information you may wish.

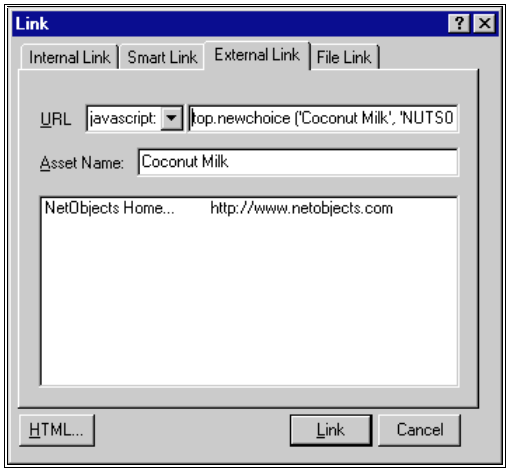

Figure 19

It would be a good idea to give the Asset a name so that you can maintain the product information using the Fusion Assets/Links tab.

The Code Generated

```
<IMG id="Picture5" HEIGHT=300 WIDTH=219
SRC="file:///H:/shopassistant/Working With/nuts_map.jpg"  BORDER=0
USEMAP="#map0" ><MAP NAME="map0"><AREA SHAPE="poly" ALT=""
COORDS="97,13,102,57,118,63,119,73,119,113,119,144,137,146,168,147,
187,143,211,135,209,13,206,5,178,2,145,1,116,4,99,11,99,13"
HREF="javascript:top.newchoice ('Coconut Milk',
'NUTS001','','',1.99,1,1,1)"></MAP>
```

Advanced Functions

The MOST flexible system

Shop@ssistant has been designed to enable you to sell products on the Internet. It is designed to make it simple to sell both unit and more complex products.

Simple products such as tins of beans have few options that might be applied to them with the exception of adding a quantity choice (an option that is provided in the review, by default).

Complex products may require the shopper to make a number of selections before they can make a purchase. A shirt is a good example of such a product.

To you and me, a shirt is a shirt, but to a shirt retailer a shirt consists of perhaps ten collar sizes, three sleeve lengths and maybe dozens of patterns and colours.

With a normal database-generated shopping system you may have needed hundreds (ten sizes times three sleeve lengths times, say, twenty colours, gives six hundred possible combinations) of entries in the database for each of these options and a page as long as your arm to display them all. However, with Shop@ssistant and a little thought about the page design this can be achieved with just three dropdown selection boxes and an Order button.

Adding Quantities

If we want to give the shopper the option of putting a quantity of the product into the basket we need to create a small form on the page.

With Fusion it is possible to treat any page as a form by ticking the 'Layout is form' in the layout properties dialog box or, if you have a number of products on the page, you could use the Fusion's 'Form Area' option to create a 'Layout Area Form'.

For our example we will use the whole page as a form since this will make the scripting simpler and will be easier to follow.

1) In Fusion, create a 'Forms Input Field' on your page into which the user may put the desired quantity. In the 'Forms Edit Field Properties' set the Name to "qty" and the Text to "1". (We assume no one will purchase 0 of an item). In our example we set the visible length to 3 and the max length to 4.

2) Now we need a button to put the product information into the basket. From the 'Form Tools' Tool bar select the Form Button and create a button on your page. Using the Form Button Properties Dialog change the Text for the button to something appropriate such as 'Add to Basket' and then set the Type dropdown to 'Button'.

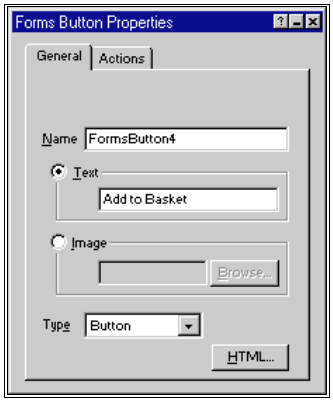

Figure 20

3) Now we need to make the button do something, so again in the Forms Button Properties dialog box click on the HTML button.

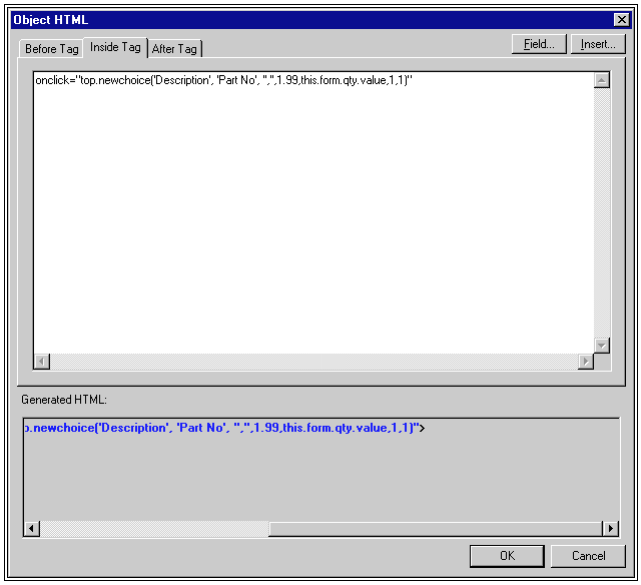

Figure 21

4) The Object HTML dialog will now be displayed. Select the 'Inside Tag' tab and in the input area, enter:

```
onclick="top.newchoice('Description','Part No','','',1.99,
this.form.qty.value,1,1)"
```

There are two special things we need to understand in the above code.

'onclick=' makes the button do something once it has been clicked; in our case it calls the function 'top.newchoice' and passes the parameters required.

To get the quantity from the "qty" input box we have replaced the quantity parameter with 'this.form.qty.value'. This has the effect of looking at the current form (this.form) and reading from the field "qty" its current value.

Variations

You may wish to validate the input quantity before you make the call to the basket. This can be done by adding some additional code to the quantity input field.

Select the input box and then click on the HTML button on its Forms Edit Field Properties dialog box.

In the Object HTML dialog box select the Inside HTML tab and enter

```
onchange="top.qty_fix (this,6,6)"
```

This will have the effect of locking the minimum quantity to 6 with a minimum quantity increment of 6. (i.e. one may purchase a minimum of six of this item, or multiples of six – maybe we're selling beer in six packs.)

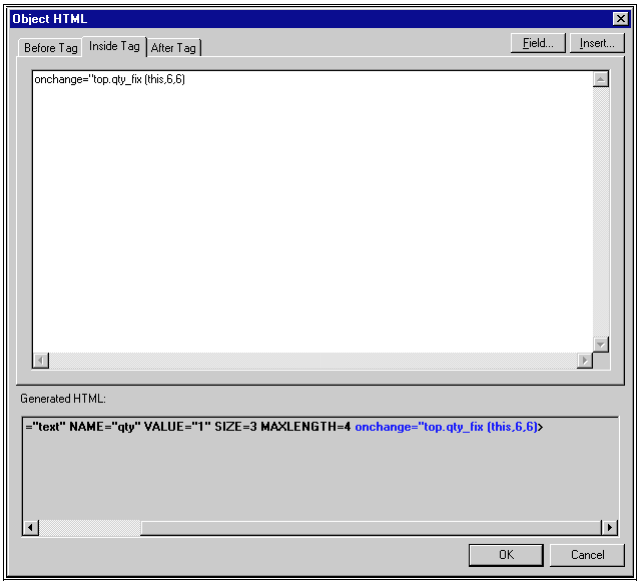

Figure 22

By using the 'onchange=' this check is only made if the shopper changes the default quantity, which you should now also set to "6" in the "qty" input fields 'Forms Edit Field Properties' dialog box.

Note: for more information on the Qty_fix and associated functions please see the Shop@ssistant developer guide.

Adding Options & Variables

Selling Shirts & Other Complex Products

In the introduction to this chapter we talked about selling complex products such as shirts. (Honestly, they are not the simplest objects to sell on the web – just you try!) The techniques used in this example could be used to sell almost anything that requires the shopper to make one or more parameter selections.

In this shirts example we are going to give the shopper the choice of selecting the colour and collar size. To ensure that they do make a choice and do not just mistakenly click the order button and get the default values, we are going to add a bit of extra code between the <HEAD> tags.

First we will create the selection boxes and the order button.

From the 'Form Tools' toolbar select the 'Form Combo Box' and create two drop down selection boxes.

The first will contain the colour selection. We will call this box Var1. (Figure 23).

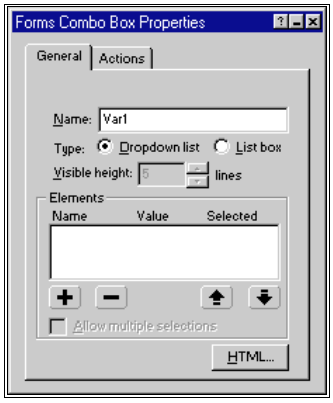

Figure 23

We now need to enter the elements for the selection. The first selection will be a message asking the shopper to make a selection, which we will make the default selection.

Note: By making the 'Please select a colour' the first selection we can test to see if the selection has been made using a property of the selection box called SelectedIndex and checking that it is not set to 0.

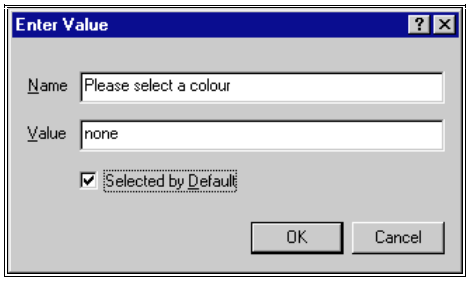

Figure 24

Now continue adding colour to the selection, The text in "Name" will be used to display the colour choice while the "Value" will be put in the basket as part of the product.

Name	Value
Flame Red	Red
Ocean Blue	Blue
Emerald Green	Green

Some example values

Next we need to create a dropdown for the collar size selection. This one we will call Var2, and again we will make the first selection a request to make a selection. (Figure 25).

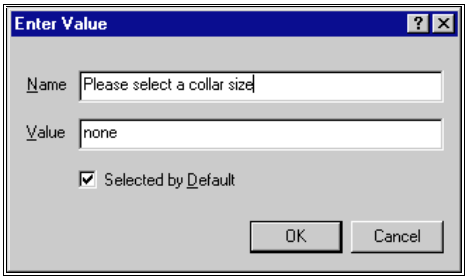

Figure 25

Now continue by adding some sizes to the selection. The text in "Name" will be used to display the size choice while the "Value" will be put in the basket as part of the product.

Name	Value
15" - 38 cm	15
16" – 40 cm	16
17" – 43 cm	17

Some example values

Finally to complete our visible page objects we will need an 'Add to basket' button.

From the 'Form Tool' bar select the 'Forms Button' and create a button on your page.

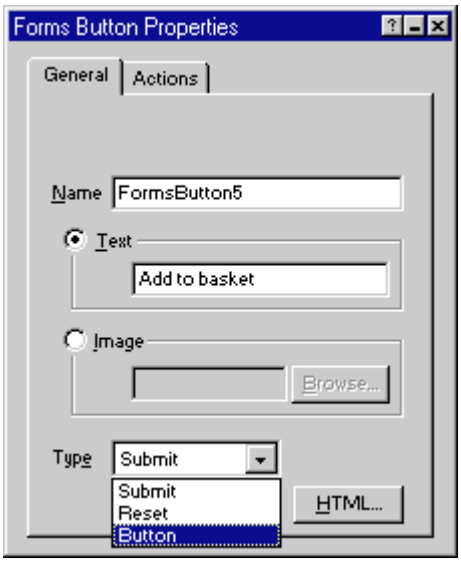

Figure 26

Change the button text to 'Add to basket' and set the button type to Button, then click on the HTML button so we can add an "onclick=" to make the button function. (Figure 27).

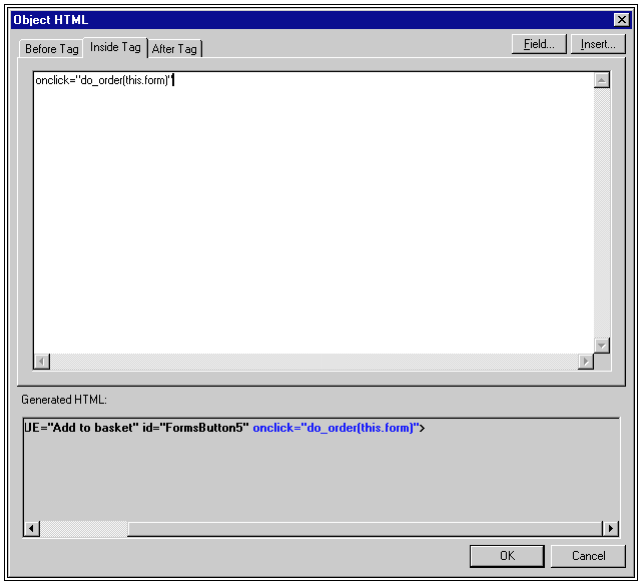

Figure 27

In the Object HTML select the Inside Tag tab and enter

```
onclick="do_order(this.form)"
```

This will make the button call a JavaScript routine called "do_order" on this page which we will later put in between the <HEAD> tags of the page. But, before we do that, we need to ensure that Fusion knows this page is a form by ticking the 'Layout is a form' checkbox in the Layout Properties dialog box.

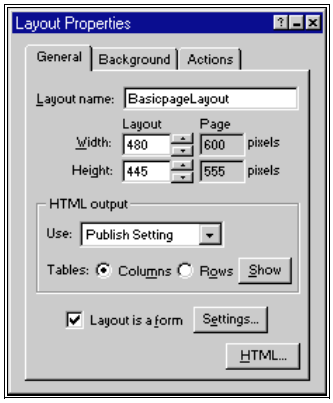

Figure 28

Now we need to create the code that will add this product to the basket. This code uses JavaScript to manipulate the variables input through the Combo Boxes we have just designed, and calls upon some routines which are built into *Shop*@ssistant and which are accessible for use in situations such as this.

The table which follows shows a suitable piece of code. This has been annotated with reference numbers and the function of the key parts of the code are explained in the notes which follow. Although this code is designed as an example, it will be found that it could cater for many of the real-life situations you may face as a web designer. There are many more examples of this sort in the Developer Demo area of the *Shop*@ssistant Demo site (SASSDemo) on the distribution CD.

Note	Script
1	<SCRIPT>
2	function do_order(form) {
3	var Message="\n"
4	var OK=true;
5	if (form.Var1.selectedIndex==0) // NO colour selected {
6	Message+="Please select a colour\n"
7	OK=false }
8	if (form.Var2.selectedIndex==0) // NO Size selected {
9	Message+="Please select a Collar Size\n"
10	OK=false }
11	if (!OK) {
12	alert (Message) }
13	// If a colour and Size is selected we can call
14	// top.newchoice() and add item to basket
15	else {
16	top.newchoice ("Shirt Description","Part_number " + top.listvalue(form.Var1) + " " + top.listvalue(form.Var2),"Colour " + top.listtext(form.Var1) ,"Size " + top.listtext(form.Var2),15.99,1,1,1) } }
17	</SCRIPT>

Notes:

1) The tag to indicate the start of the script.
2) The name of the function we have created, followed by the (form) parameter which indicates where the input to the function is taken from.
3) Declares a message variable called “Message” and sets this to a blank.
4) Declares a check variable (OK) and sets this to “true”.
5) Checks IF a selection has been made from Combo Box “Var1” (in which case the value of Selected Index would be greater than 0).
6) Changes the text in “Message” to reflect the fact that no colour choice has been made.
7) Changes the check variable value to “false” to indicate that the input evaluated is incorrect.
8) Checks IF a selection has been made from Combo Box “Var2” (in which case the value of Selected Index would be greater than 0).
9) Changes the text in “Message” to reflect the fact that no size choice has been made.
10) Changes the check variable value to “false” to indicate that the input evaluated is incorrect.
11) Checks if the input has been completed correctly by looking at the value of the check variable “OK”. If the value of OK is “false” (meaning that one of the input fields has not been correctly completed), then proceeds to line 18 and prints the error warning message. Otherwise (if OK is “true”) skips to the line starting “else”, referenced 16.
12) Prints the value of the warning message held in the variable “Message” in an Alert box.
13) This is a comment for the programmer’s guidance.
14) This is a comment for the programmer’s guidance.

15) Marks the start of the code to be executed if the condition checked in line 16 is not true.

16) This line makes the call to the basket (top.newchoice) and passes the information about the product selected for display in the Review page.

17) Marks the end of the script.

Putting the code into your page

Ensure that you are working in the layout region of your page by clicking in any of the blank spaces in the layout region. On the "General" tab of the Layout Properties click on the "HTML" button to open the Page HTML dialog. The text of the script we have created should be inserted on the page tabbed "Between Head Tags". If you have created the script in a text editor (such as TextPad, supplied on the distribution CD) then you can simply 'cut' the text from your text editor and 'paste' it into the page here. Otherwise you may type the script directly into the page. Either way, you should end up with a page that looks as follows, Figure 29.

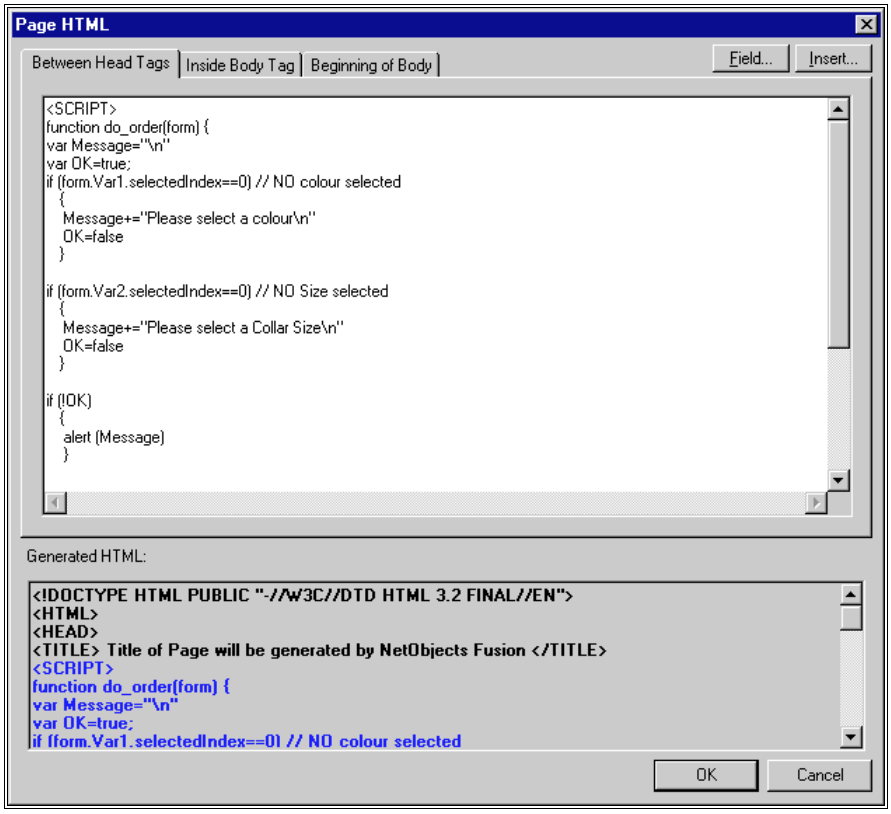

Figure 29

Once complete, this window may be closed and your page, when opened under the *Shop*@ssistant shop system will operate as planned, allowing you to select a shirt with your choice of colour and collar size. The easiest way to see your page running is to copy this page to the /pages folder of your site and call it 'index,html'. It will then be the first page loaded after the system itself has loaded.

Built-in Functions (Shop@ssistant)

Shop@ssistant provides a number of functions which can be of great use to the site builder. Please check the system documentation and the demonstration site on the distribution CD for more information. The Developer Demo Area of the demo store is a good place to start to get a better understanding of the flexibility and power inherent in *Shop*@ssistant.