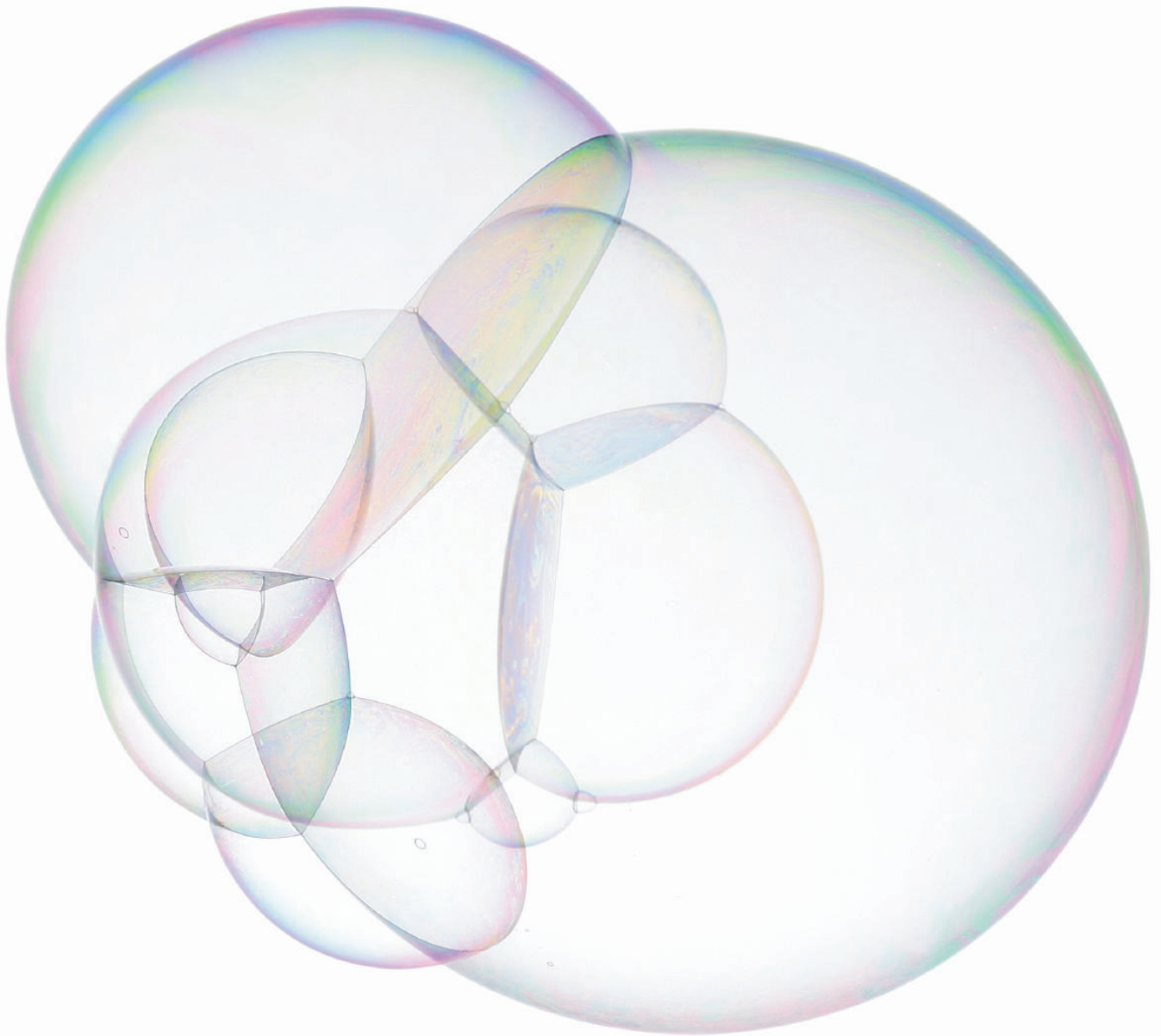


webShaper e-commerce

Template Design & Integration Guide



Designing for webShaper e-commerce

This section will explain the design details when designing for e-commerce. It's divided into:

- i. Plug-In Components**
- ii. Designing for Dynamic Pages**
- iii. Template Design Guidelines**

i. Plug-In Components

This section is to highlight the possible things that can be put into the template files, or static pages (not all):

- Search Box Form
- Category Listing Component
- RSS Link
- Product Map
- News Component
- Newsletter Form
- New Products Component *
- Best Sellers Component *
- Specials Component *
- Featured Products Component *
- Currency Component
- Price Range Component
- Browse by Custom Field
- Manufacturers List Component
- Themes List Component
- Cart Component

* - Items which can be inserted in both template files and static files, the tags are case sensitive.

Search Box Form

The search box code below, will post queries to searchProd.asp. Please note that you may use a button, defined in the skin.xml `<ws_button id="btnSearch" />`.

```
<form name="formSearch" method="post" action="/webshaper/store/searchProd.asp"
class="form" >

    <div style="float:left; margin:3 0 0 0;" >
        <input class="textbox" id="searchText" size="18"
            name="searchText" value="" onblur="if(this.value=='') this.value='';"
onfocus="if(this.value=='Search Product') this.value='';" />

    </div>

    <div style="float:left" >
        <ws_button id="btnSearch" />
    </div>

    <div style="float:left;margin:5 10 0 10;" >
        <a href="/webshaper/store/advancedSearch.asp">Advanced</a>
    </div>

    <div class="clear" />
</form>
```

Category Listing Component

Category listing will list down categories level using a tree view structure. To call up the component, enter `<categoryListing/>` in either one of the template files.

Now for some introduction on components. Each component has it's own XML data file, and also an XSL file. For category listing, it's categoryList.xml and categoryList.xsl. Then you can further edit the configuration inside skin.xml.

The XML data file stores the list of all categories, while the xsl file is used to style the component.

However, there are some css class hooks into the component, so you don't need to open up the XSL file to style it.

Below is the HTML structure for category listing, under categoryList.xsl.

```

<div class="categoryList" >
  <div class="categoryListHeader" >
    [ws_component/header/text]
  </div>
  <div class="categoryListBody" >
    <ul id="categoryList_ul_0" >
      <li><a href="..." >category 1</a></li>
      <li><a href="..." >category 2</a></li>
      <ul>
        <li><a href="..." >category 2.1</a></li>
        <li><a href="..." >category 2.2</a></li>
      </ul>
    </ul>
  </div>
  <div class="categoryListFooter" >
  </div>
</div>

```

categoryListing	
CSS class	Description
categoryList	div wrapper
categoryListHeader	div header
categoryListBody	div body
categoryListFooter	div footer

Besides that, under the skin.xml section, you can edit the following:

- i. Component header text
- ii. Category Depth Level

RSS Link

webShaper E-commerce RSS feed file contains news listing and also Featured Products List that is updated every time user logs off from the store.

The RSS file is stored at folder /webshaper/store/index.xml

To link to it, just add something like below:

```

<a href="/webshaper/store/index.xml" >&#160;RSS Feed</a>

```

Product Map

The Product Map page list down all categories and products under the site. It's meant to help search engine robots to access the link to these links immediately from the website. Therefore, it's advisable to expose out this link on the template pages.

Example code:

```
<a href="/webshaper/store/productMap.asp" >Product Map</a>
```

News Component

The News Component is a list of news title and it's date that can be inserted inside the template. To call up the component, use the code `<newsComponentList/>`

HTML structure inside newsComponent.xsl

```
<div class="newsComponent" >
  <div class="newsComponentHeader" >
  </div>
  <div class="newsComponentBody" >
    <ul>
      <li><a href="...">news title 1<br/>12 Dec 2006</a></li>
      <li><a href="...">news title 2<br/>12 Dec 2006</a></li>
    </ul>
  </div>
  <div class="newsComponentFooter" >
  </div>
</div>
```

newsComponentList	
CSS class	Description
newsComponent	div wrapper for component
newsComponentHeader	div header
newsComponentBody	div body
newsComponentFooter	div footer

Under skin.xml, you can edit:

i. Component header text

Newsletter Form

Normally, the Newsletter Form is exposed out in the template pages, so that visitors can immediately sign up for latest news or updates via email.

The HTML example code for the newsletter form is as below:

```
<form name="FormSubscribe" method="post"
action="/webshaper/store/newsletterSubscribe.asp" onsubmit="return CheckFormSubscribe()"
>

    <input type="hidden" name="fp" value="1" />

    <div>

        <label for="custName" >Name:</label><br/>
        <input id="custName" size="20" name="custName" value="" class="textbox"
/><font color="red" >*</font><br/>

        <label for="email" >Email:</label><br/>
        <input id="email" size="20" name="email" value="" class="textbox" /><font
color="red">*</font><br/>

        <label for="mobilePhone" >Mobile: </label><br/>
        <input id="mobilePhone" size="20" name="mobilePhone" value=""
class="textbox" />

        <ws_button id="btnSubscribe" />

    </div>
</form>

<!--
for the form checking, please add this to your template file:

<script language="JavaScript" src="/webshaper/store/0js/template.js"
type="text/javascript"></script>

-->
```

New Products Component, Best Sellers Component, Specials Component

This component displays a list of products on templates files, and also statics pages. It is called up using <newProductsComponentList/>, <bestSellersComponentList/>, <specialsComponentList/>.

You can configure its display either horizontally, or vertically aligned at the Config section.

The products list are actually stored in cached XML files, which cuts down the time to display them on the template.

HTML example code <newProductsComponentList/>:

```

<div class="newProductsComponent" >
    <div class="newProductsComponentHeader" >
        <div class="newProductsComponentHeaderText" >
            Header Text
        </div>
    </div>

    <div class="newProductsComponentBody" >

        <!--javascript-->
        <script language="javascript"
src="/webshaper/store/0js/newProductsComponent.js" type="text/javascript"></script>

<form name="FormNewProductsComponent" method="post" action="" >
<input type="hidden" name="PKProductItem" value="" />
<input type="hidden" name="TotalProduct" value="1" />
<input type="hidden" name="iPKProduct0" value="" />
<input type="hidden" name="qtyProduct0" value="1" />

<table class="newProductsComponentGrid" >
    <tr>
        <td>
            [Product HTML]
        </td>
    </tr>
</table>

        </form>

<div style="text-align:left;width:;margin:0 0 0 6" >
<a href="/webshaper/store/newProducts.asp" >&gt; More Items...</a>
</div>

        </div>
    <div class="newProductsComponentFooter" >

    </div>

</div>

```

[Product HTML]

HTML structure for product inside a component.

```
<div >
  <div >
    <a
href="/webshaper/store/viewProd.asp?pkProductItem={$pkProductItem}" >img
src="{picURL}" border="0" alt="{prodName}" /></a>
    </div>
    <div >
      <xsl:value-of select="{prodName}" />
    </div>
    <div>

      [Price display]
    </div>

    <a
href="javascript:InsertCartNPcomp (' {$pkProduct} ', ' {$pkProductItem} ')" >Add to
Cart</a>
    &#160;
    <a
href="/webshaper/store/viewProd.asp?pkProductItem={$pkProductItem}" >More
Info</a>
  </div>
```

newProductsComponent	
CSS class	Description
newProductsComponent	div wrapper for component
newProductsComponentHeader	div header
newProductsComponentHeaderText	Div header text
newProductsComponentBody	div body
newProductsComponentFooter	div footer

Note: For Specials component, the convention will be:
“specialsComponent” + header, body, footer, grid. The same applies to “bestSellersComponent”.

Featured Products Component

This component can only be displayed on the static page, usually index.html. It's called up using `<featuredProductsList/>`.

HTML example code:

```
<div class="featuredProducts" >
  <div class="featuredProductsHeader" >
    <div class="featuredProductsHeaderText" >
      [Header Text]
    </div>
  </div>
  <div class="featuredProductsBody" >
    <form name="FormFeaturedProducts" method="post" action=""
class="form">
      <input type="hidden" name="PKProductItem" value="" />
      <input type="hidden" name="TotalProduct" value="1" />
      <input type="hidden" name="iPKProduct0" value="" />
      <input type="hidden" name="qtyProduct0" value="1" />

      <table class="featuredProductsGrid" >
        <tr>
          <td>
            [Product Item HTML: Please view source to see more
details on the CSS class.]
          </td>
        </tr>
      </table>

    </form>
  </div>
  <div class="featuredProductsFooter" >
  </div>
</div>
```

Featured Products Component	
CSS class	Description
featuredProducts	div wrapper for component
featuredProductsHeader	div header
featuredProductsHeaderText	div header text
featuredProductsBody	div body
featuredProductsFooter	div footer

Currency Component

The Currency Component is used to show a list of currencies available, in a drop down list. It's called up using `<currencyList/>`.

Please add this code in `template.html` body onload event:

```
<body onload="initCurrency();" >
```

The javascript function 'initCurrency()' is defined in `/webshaper/store/0js/template.js` to set the currency selected on the currency component, every time the web site loads.

HTML example code:

```
<div class="currencyComponent" >
  <div class="currencyComponentHeader" >
    <div class="currencyComponentHeaderText" >
      [Header Text]
    </div>
  </div>
  <div class="currencyComponentBody" >
    <form name="formCurrency" method="post" >
      <select name="currency" onchange="selectCurrency(this)"
class="dropdownlist">
        <option value="[currCode]" >
          [currName]
        </option>
        <option value="[currCode]" >
          [currName]
        </option>
      </select>
    </form>
  </div>
  <div class="currencyComponentFooter" >
  </div>
</div>
```

Currency Component	
CSS class	Description
currencyComponent	div wrapper for component
currencyComponentHeader	div header
currencyComponentHeaderText	div header text
currencyComponentBody	div body
featuredProductsFooter	div footer

Price Range Component

The Price Range Component is used to show a few pricing tier, which allows customer to browse by price for the whole store. It's called up using `<priceRangeList/>`.

HTML example code:

```
<div class="priceRangeComponent">
  <div class="priceRangeComponentHeader">
    <div class="priceRangeComponentHeaderText">
      [Header Text]
    </div>
  </div>
  <div class="priceRangeComponentBody">
    <ul>
      <li><a
href="/webshaper/store/advancedSearch.asp?relation=morethan&priceTo=350&F
lagProcess=1&sort=priceLH">
      More Than MYR 350</a></li>
      <li><a
href="/webshaper/store/advancedSearch.asp?relation=lessthan&priceTo=350&F
lagProcess=1&sort=priceLH">
      Less Than MYR 350</a></li>
    </ul>
  </div>
  <div class="priceRangeComponentFooter"></div>
</div>
```

Price Range Component	
CSS class	Description
priceRangeComponent	div wrapper for component
priceRangeComponentHeader	div header
priceRangeComponentHeaderText	div header text
priceRangeComponentBody	div body
priceRangeComponentFooter	div footer

Browse by Custom Field

This component is used to enable browsing by custom field defined in the store. You can put as many custom fields as you want on any templates files. It's called up using:

```
<customFieldsList code="[code of the custom field]" title="[Text to enter e.g. `Shop by Color`]" />
```

HTML example code:

```
<div class="customFieldsList" >
  <div class="customFieldsListHeader" >
    <div class="customFieldsListHeaderText" >

      </div>
    </div>
  <div class="customFieldsListBody" >

    <ul>
      <li><a
href="/webshaper/store/browseByCustomField.asp?pkCSF=[code1]&search=[escapeVa
lue1]" >[custom field value 1]</a>
      </li>
      <li><a
href="/webshaper/store/browseByCustomField.asp?pkCSF=[code2]&search=[escapeVa
lue2]" >[custom field value 2]</a>
      </li>
    </ul>

    </div>
  <div class="customFieldsListFooter" >

  </div>
</div>
```

Browse by Custom Field	
CSS class	Description
customFieldsList	div wrapper for component
customFieldsListHeader	div header
customFieldsListHeaderText	div header text
customFieldsListBody	div body
customFieldsListFooter	div footer

Manufacturers list component, Themes list component

Both these components can be used to group product items under a few manufacturers, or themes. They display the list of manufacturers and themes. It's called up using:

```
<mfcComponentList/>  
<themeComponentList/>
```

HTML example code:

```
<div class="mfcComponent">  
  <div class="mfcComponentHeader">  
    <div class="mfcComponentHeaderText">  
      [header text]  
    </div>  
  </div>  
  <div class="mfcComponentBody">  
    <ul>  
      <li><a href="/webshaper/store/manufacturer.asp?mfc=160">Acer</a></li>  
      <li><a href="/webshaper/store/manufacturer.asp?mfc=159">HP</a></li>  
    </ul>  
  </div>  
  <div class="mfcComponentFooter"></div>  
</div>
```

Note: For themes, use “thmComponent” + header, headerText, Body, Footer.

Manufacturers List Component, Themes List Component	
CSS class	Description
mfcComponent	div wrapper for component
mfcComponentHeader	div header
mfcComponentHeaderText	div header text
mfcComponentBody	div body
mfcComponentFooter	div footer

Cart Component

The cart component is a mini basket which shows the list of items in customer's basket. It's called up using: `<cartComponentList/>`

HTML example code:

```
<div class="cartComponent">
  <div class="cartComponentHeader">
    <div class="cartComponentHeaderText">
      [Header text: e.g. "Your Cart"]
    </div>
  </div>
  <div class="cartComponentBody">
    <ul>
      <li>1 x Product 1</li>
    </ul>
    <div>
      <a href="/webshaper/store/viewCart.asp">View / Edit Cart</a>
      <a href="/webshaper/store/registerLogin.asp">Checkout</a></div>
    Subtotal: MYR 38.50</div>
  <div class="cartComponentFooter"></div>
</div>
```

Cart Component	
CSS class	Description
cartComponent	div wrapper for component
cartComponentHeader	div header
cartComponentHeaderText	div header text
cartComponentBody	div body
cartComponentFooter	div footer

ii. Designing for Dynamic Pages

Dynamic pages consist of the following pages (stored under /webshaper/store):

- addressBilling.asp
- addressBook.asp
- addressRegister.asp
- addressShipping.asp
- advancedSearch.asp
- bestSellers.asp
- browseByCustomField.asp
- cancelpayment.asp
- changePassword.asp
- contactUs.asp
- createAccount.asp
- cust_createNewShipAddr.asp
- cust_editShipAddr.asp
- cust_modifyInfo.asp
- cust_modifyShipInfo.asp
- custHome.asp
- custOrderList.asp
- emailFriend.asp
- forgetPassword.asp
- manufacturer.asp
- message.asp
- newProducts.asp
- news.asp
- newsletter_subscribe.asp
- newsletter_unsubscribe.asp
- orderSummary.asp
- payment.asp
- ProductMap.asp
- registerLogin.asp
- searchProd.asp
- shipping.asp
- sign_in.asp
- specials.asp
- thankyou.asp
- theme.asp
- verifyAccount.asp
- viewCart.asp
- viewCat.asp
- viewProd.asp

These pages do not generate the HTML directly to the browser. Instead, it generates XML data, which are combined with their respective XSL, to generate the content part of the website, which is then merged with the template files.

There are 3 ways to change the appearance of the dynamic pages.

- i. Using CSS (Cascading Style Sheet) to style each dynamic page.
- ii. Editing skin.xml to edit sections of the dynamic pages (e.g. header, sub header, and includes leftbar, rightbar, header, footer)
- iii. Editing their XSL file (/webshaper/template/components). This gives you **full power** to change anything you want. However this requires advanced HTML knowledge and familiarity with XSL. The easiest way is to open the corresponding XSL file. For example, to change viewCat.asp, just open viewCat.xsl and modify the codes from there.

General HTML framework for dynamic pages

To make things standardized, each dynamic page has a general HTML code wrapped around.

```
<div class="pageContent" >
  <div class="pageHeader" >
    [Editable in skin.xml]
  </div>

  <div class="pageSubHeader" >
    [Editable in skin.xml]
  </div>

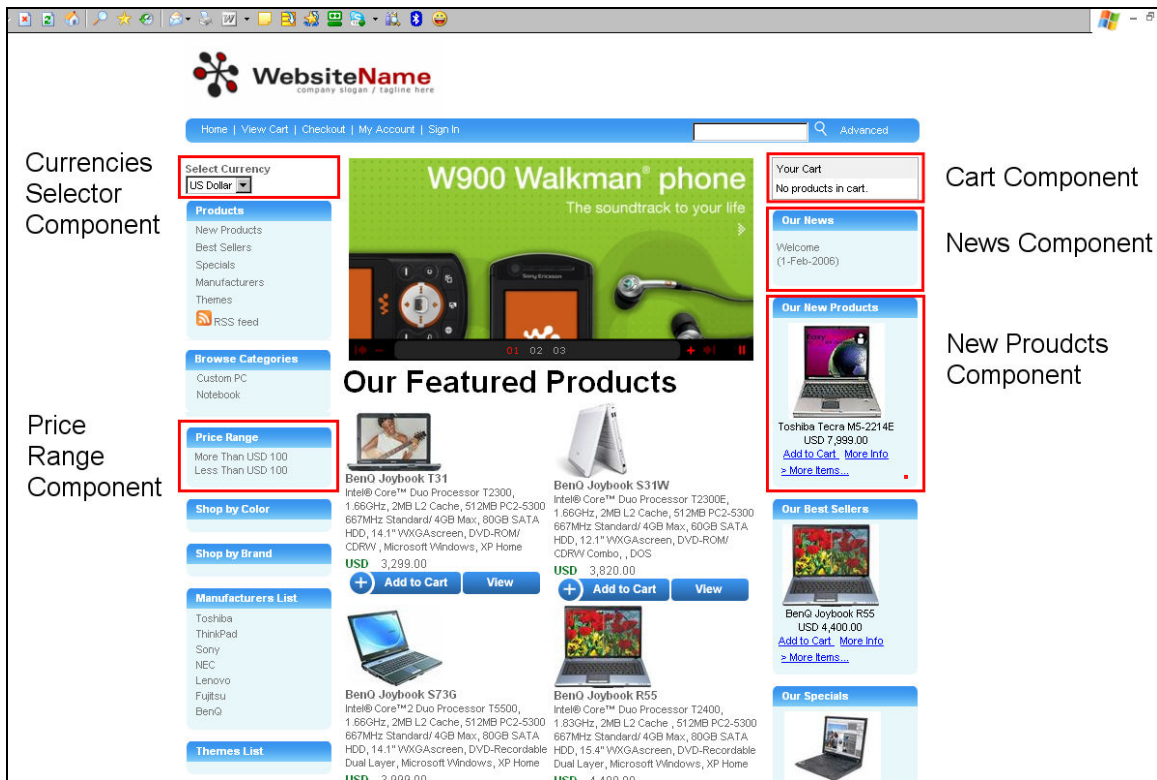
  <div class="pageBody" >
    [Where dynamic content starts...]
  </div>
</div>
```

Therefore, you can use CSS to position your dynamic section accurately.

iii. Template Design Guidelines

- 1) Use Text for Any Links whenever possible
 - a. Keep in mind purpose of template is to empower your customers to make changes whenever possible, or on area which they most likely to change.
 - b. Component designed as images is hard to change. Ask questions like: “What customers want to change?” Some obvious are logo, static page links etc.

- 2) Every template must include the below “component”, which are functionalities provided by webShaper e-commerce by default.
 - a. News
 - b. New Products
 - c. Best Sellers
 - d. Specials
 - e. Newsletter
 - f. Manufacturers etc



Refer to standard demo store at <http://e-commerce.neowave.com.my/>

- 3) Reserve a place for “Your Company Logo”, sample as below.



Sample 1



Sample 2

4) Buttons (refer to Skin.xml)

Button Name	Button Images (Standard)
btnAddToCart.gif	
btnBack.gif	
btnCheckout.gif	
btnClear.gif	
btnCreateAccount.gif	
btnNext.gif	
btnPagingNext.gif	
btnPagingPrev.gif	
btnSearch1.gif	
btnSignIn.gif	
btnSubmit.gif	
btnSubscribe.gif	
btnUnsubscribe.gif	
btnUpdate.gif	
btnUpdateCart.gif	
btnView.gif	
button_tell_a_friend.gif	

The information contained in this document represents the current view of Neowave Sdn Bhd on the issues discussed as of the date of publication. Because Neowave must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Neowave, and Neowave cannot guarantee the accuracy of any information presented after the date of publication.

This overview is for informational purposes only. Neowave MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. Use it at your own risk.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Neowave Sdn Bhd.

Neowave may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Neowave, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Neowave and webShaper are either registered trademarks or trademarks of Neowave Sdn Bhd in Malaysia and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.