



ProStores Designer's Reference

Version 8.1



© 2007 ProStores Inc.
ProStores, and the ProStores
logo are trademarks of
ProStores Inc. All other
products mentioned are
trademarks of their respective
companies.

Table of Contents

Chapter 1: Introduction.....	4
The Importance of Store Design	4
Additional Documents.....	4
Chapter 2: Understanding Store Design.....	5
How Data is Displayed.....	5
Understanding Templates.....	7
Understanding SSML Tags	8
Understanding Business Objects.....	9
Chapter 3: Integrating an Existing Site with ProStores.....	10
Transferring the Look and Feel of Your Existing Site to Your Storefront....	10
Choosing a Theme.....	10
Customizing a Theme.....	11
Copying Content From Another Web Site	12
Linking to and from Your Storefront	14
Create Custom Links to Your Store	14
Link Manager	16
Chapter 4: Working with ProStores Store Design	18
Edit Page Template	18
Managing Your Work	19
Set a Baseline	20
Revert to Baseline	21
Revert to Original Version	21
Publish Pages	21
Common Tasks.....	22
Selecting Content	22
Adding Text	25
Adding Links.....	25
Adding Images	27
Display Store Data	28
Building Lists.....	29
Using Conditions.....	30
Saving and Undoing Your Work.....	32
Advanced Tasks	32
Validate Page Template	32
Import and Replace Page Template.....	33
Export Page Template	33
Using Template Edit Modes.....	34
Basic Editor	37
Using the Advanced Editor	37
Using the Template Source Editor	42
Chapter 5: Using Store Versions.....	43
Understanding Store Versions.....	43
Store Versions	44
Edit Language/Locale	46

Edit Rules	46
Edit Status	49
Chapter 6: Advanced Design Concepts	50
Working with Images	50
Uploading Images	50
Using and Updating Store Images	53
Using Product (Catalog) Images	53
Using an Image in Place of the "Submit" Button	54
Customizing Tag Images	54
Working with Files	55
Uploading Files	55
Creating a Splash Page	55
Cascading Style Sheets	56
JavaScript	57
Multimedia Files	57
Managing Files	57
Omitting the Header and Footer	58
Customizing the Display of the Category List	59
Using Multiple Templates	59
Where You Can Use Multiple Templates	59
How to Create and Use Multiple Templates	60
User Defined Templates	61
Frequently Asked Questions: Store Design	61
Chapter 7: Code Snippets	66
Support Multiple Featured Products and Categories	66
Support Advanced Fonts and Colors	67
Adding Multiple Products to a Cart	69
Quick Search Form	69
UPS Tracking Link	69
Collapsible Category Lists	70
Product Price Formatting	70
Hide Attribute Information When Not In Stock	71
Order Extension Information on the Invoice Email	72
Suppressing A Shipping Method	72
Subcategory Drill-Down on Catalog List	72
Comparison Category List	74
Display Related Products on Cart Template	75
Show Only Attributes That Are On Sale	75
Total Number of Items in a Cart	76
Chapter 8: Testing Your Design	77
Equipment and Software to Test	77
How to Test	77
Testing Store Information Templates	77
Testing the Product Catalog Templates	78
Testing the Checkout Templates	78
Testing Membership and Subscription Templates	78
Testing Customer Service Templates	79
Troubleshooting	79
Troubleshooting Validation Errors	79

Troubleshooting Images	79
Troubleshooting Pages	80
Troubleshooting Search Results.....	81
Appendix A: Glossary.....	83
Appendix B: Template Reference	87
Appendix C: Tag Reference.....	108
Appendix D: Tag Mappings	137
Index	175

Chapter 1: Introduction

The Designer's Reference is designed to provide you with all the information you need to create the perfect look and feel for your store. This guide covers all phases of the design process, from using the automated wizards, to advanced custom store design using JavaScript and Cascading Style Sheets. In addition, this guide contains detailed ProStores Tag and Template reference.

The Importance of Store Design

The design of your store is critical when customers are deciding whether or not to purchase products from your store. The design of your store affects:

- ◆ How customers perceive your credibility. A design that looks unprofessional or outdated may cause a customer to look elsewhere for products.
- ◆ Whether or not a customer will complete a purchase. A design that is hard to use may cause a customer to cancel their purchase.
- ◆ A customer's willingness to trust you with payment information. Again, an unprofessional design may cause a customer to think you don't take security concerns seriously.
- ◆ Whether or not a customer will return to your store. If the design makes it difficult for a customer to find what they are looking for or check out, they may be less likely to return to your store.

The themes and layouts developed by ProStores take all these factors into consideration. Before making modifications, or developing your own design, we recommend that you become familiar with web design principles before continuing.

Additional Documents

Please see the following documents for additional information:

- ◆ The Online Guide available in the ProStores application.
- ◆ ProStores Store Administration Users Guide: this may be downloaded from the Online Guide start page.
- ◆ ProStores SSML Object Reference: this may be downloaded from the Online Guide start page.

Chapter 2: Understanding Store Design

Starting with the storefront through order confirmation, templates define your store's look and feel. One template defines your store's navigation, another defines how your product catalog is displayed, and another defines how the e-mails your store sends will appear.

ProStores is shipped with a default set containing dozens of templates that cover every phase of the shopping process. However, you can modify these templates to provide your own custom look and feel.

In addition, ProStores supports the uses of additional pages and files (like non-store generated HTML pages, JavaScript and CSS) that allow for unlimited possibilities when developing the design of your store.

This reference will provide you with the information you need to create the perfect store design.

How Data is Displayed

Usually, web pages are made up of just a few components, such as HTML Code, text content, and images. More advanced pages might also have a style sheet, JavaScript and other multimedia. Regardless of the specific elements, once you have created a page, it will always appear the same every time it is viewed. These are considered "static" pages.

ProStores, however, combines standard HTML elements with ProStores Markup Language tags (SSML) and "Business Objects." These tags and objects display data from your store's database and generate pages based on your catalog, customer information, search information, store settings, etc. These are considered "dynamic" pages.

For example, if you were using static pages to create your store and wanted to create a table that would display a list of your products, it might look something like this:

```
<table>
  <tr>
    <td>Sleeping Bag</td>
    <td>$300.00</td>
  </tr>
  <tr>
    <td>Tent</td>
    <td>$150.00</td>
  </tr>
  <tr>
    <td>Outdoor Cooking Set</td>
    <td>$750.00</td>
  </tr>
</table>
```

This would display the product information for the three products listed on this page. Once you've created this page, the only way to update it for your new products, or to change pricing information, is to open the page and add a new row for each product.

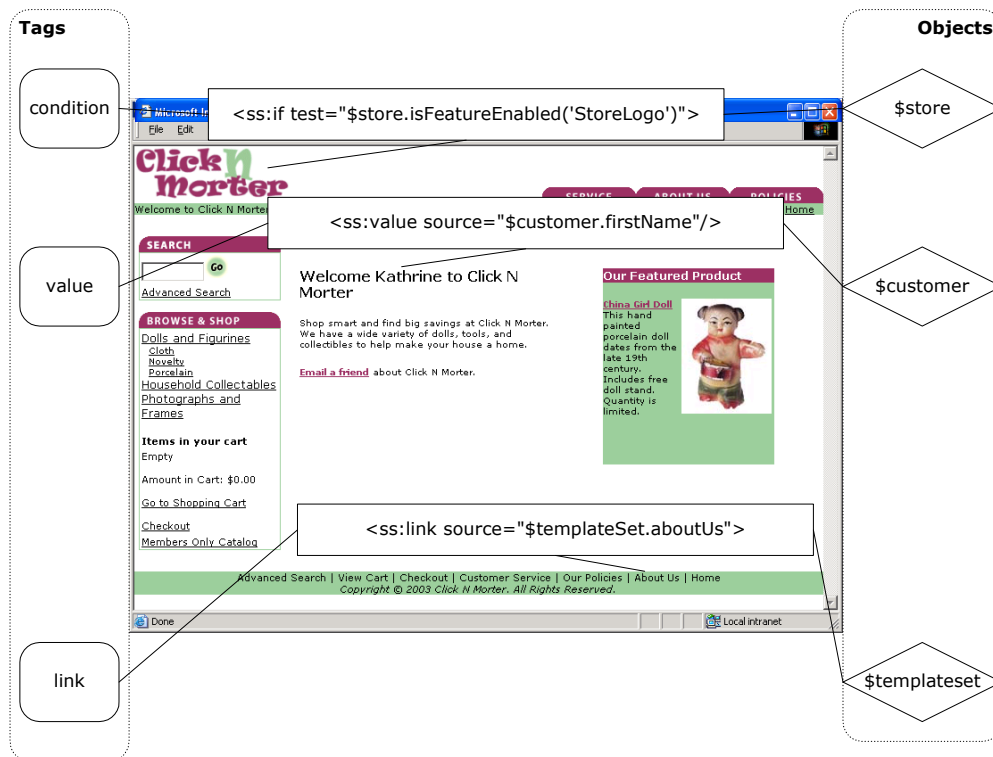
Using a ProStores template however, you would still code a table, but instead of having one row for each product, you would create code that defines how you want a product row to appear. You would then use SSML tags and objects in place of the actual product information. The next image shows an example of a template - SSML tags begin with <ss:.

```
<table>
  <ss:foreach item="listing" within="$catalog.results">
    <ss:foreach item="product" within="$listing.products">
      <ss:sortby item="product" />
      <tr>
        <td><ss:value source="$product.name" /></td>
        <td><ss:value source="$product.price" /></td>
      </tr>
    </ss:foreach>
  </ss:foreach>
</table>
```

- ◆ The <ss:foreach> and </ss:foreach> tags tell the page to repeat the information between those tags for each product.
- ◆ The <ss:value source="\$product.name" /> tag and object displays the product name.
- ◆ The <ss:value source="\$product.price" /> tag and object displays the price of the product, formatted as currency.

Now, whenever you add a new product, that product will appear on this page. Or, if you change a product's price, that new price will appear the next time the page is viewed.

ProStores templates also work with your store's information when generating pages. For example, if you set the maximum search results per page to 10, then "next" and "back" links are automatically added if your customer's search results generate more than 10 products.

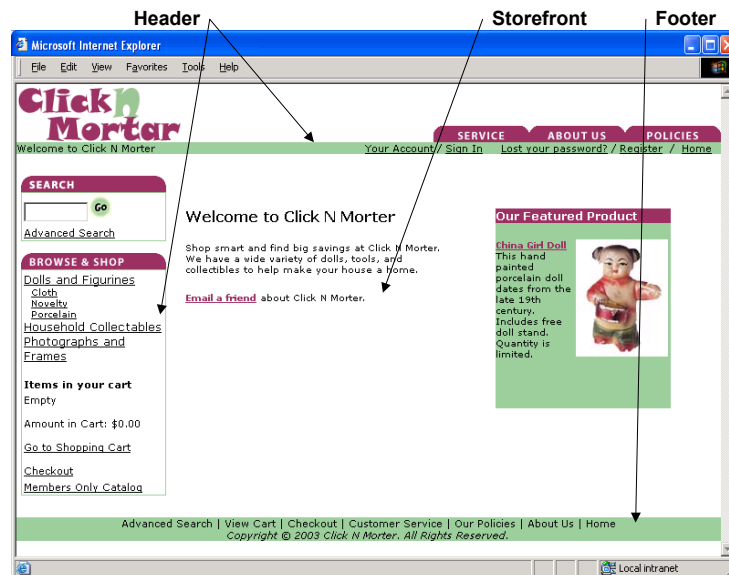


How store information is used to build templates

Understanding Templates

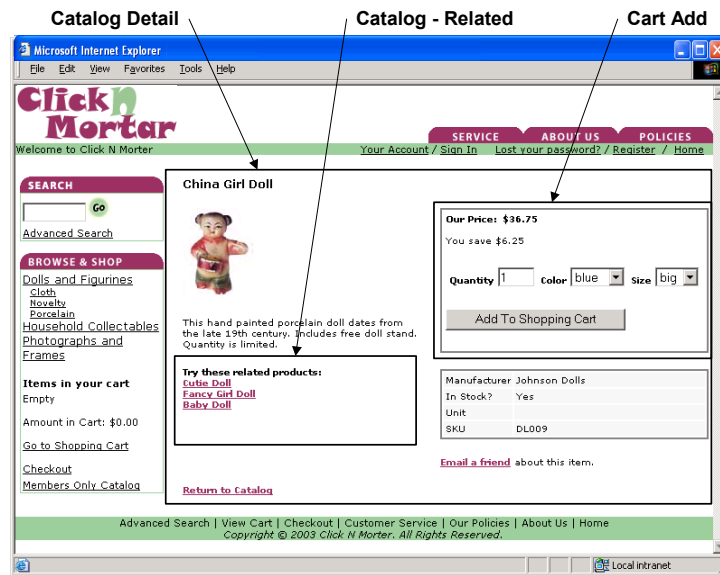
Unlike standard HTML pages, ProStores template pages are “dynamic”, meaning they are built as the customer requests them. In addition, what appears to be a single page when viewed in the browser is actually at least three files put together into a page - the header template, footer template, and then a body template.

This is done for two reasons: to simplify the design of your store, and to offer maximum customization of your store. For example, by changing the navigation on your header template, you have changed the navigation for your entire store.



Templates used to build your store's opening page.

However, some pages are built using even more templates. For example, the Catalog Detail template can contain both the Related Products and Cart Add templates.



Templates required to build the Catalog Detail page.

To prevent the Header and Footer templates from displaying on a template, simply add a `<HTML>` tag as the first item on the appropriate template, and a `</HTML>` template as the last item. This allows you to achieve custom designs for individual templates (such as giving the home page a different look than the rest of your templates).

Understanding SSML Tags

ProStores Markup Language (SSML) Tags are the equivalent of a markup language such as Hypertext Markup Language (HTML) or Standard Generalized Markup Language (SGML), and define the type of information that will be presented (for example, a link is created using the `<ss:link/>` tag). Tags can:

- ◆ Display information from the store (`<ss:value/>`, `<ss:image/>`).
- ◆ Create form fields (`<ss:edit/>`, `<ss:select/>`, `<ss:password/>`, `<ss:radio/>`).
- ◆ Apply formatting (`<ss:font/>`).
- ◆ Create links (`<ss:link/>`).
- ◆ Display information based on test conditions (`<ss:choose/>`, `<ss:if/>`).
- ◆ Create and sort lists (`<ss:foreach/>`, `<ss:sortby/>`).

In addition, SSML includes more advanced tags that can assist in the creation of complex store functionality and design.

The following are examples of how SSML tags are constructed:

```
<ss:edit source="$customer.FirstName" size="20"/>  
(1) (2) (4) (3) (4) (6)
```

```
<ss:link source="$templateSet.aboutUs">About Us</ss:link>  
(1) (2) (4) (3) (5) (6)
```

- 1: <ss:/> defines this as a ProStores tag.
- 2: The tag type. Generally indicates what this tag will resolve to on the storefront.
- 3: The Object Expression. Defines where the data is coming from, or the data to which input will be posted. Object expressions always start with the "\$" character.
- 4: Tag attributes that further define the tag.
- 5: The content to appear between the opening and closing tags. Not all tags have content.
- 6: The closing slash or tag.

Understanding Business Objects

ProStores Business Objects provide direct access to the information in a your database, and combined with HTML or the ProStores tags, display or collect information in the storefront. For example, <ss:value source="\$customer.firstName"/> displays the first name of the customer (the <ss:value/> tag displays a value, and the \$customer.firstName object expression tells the tag what item to display).

Business Objects provide access to information such as:

- ◆ Product Information: \$product.description, \$product.name, \$product.thumbnailImageName.
- ◆ Search Options: \$catalog.categoryList, \$catalog.keyword.
- ◆ System Information: \$system.daysTill('Christmas') .
- ◆ URL Paths: \$templateSet.images['filename.gif'], \$templateSet.AboutUs.

Chapter 3: Integrating an Existing Site with ProStores

While in many cases, ProStores' pre-designed pages and extensive capabilities make maintaining a separate web site and web store unnecessary, you may prefer to keep your site. If so, you'll want to design your store so that it flows seamlessly with your existing site. This section provides design tips and instructions for linking your site with your store.

Transferring the Look and Feel of Your Existing Site to Your Storefront

The first step in transferring the look and feel of your existing site to your storefront is to identify those elements that make up your site's design.

- ◆ **Text Fonts and Colors:** What fonts and colors are you using for your web page text? Make a note of these, and when you use the ProStores tools, make the selections that match your existing site.
- ◆ **Page Layout:** How would you describe the layout of your store? Does your site's navigation run along the top or one side? If so, you'll want to modify your storefront's theme to use the same navigation.
- ◆ **Images:** While you may have many images on your page, identify those that are consistent across most pages - those images are the ones to use in your store.

Once you've identified your design elements, you're ready to transfer that design to your storefront.

Choosing a Theme

If you have not already selected a theme, you can select one that matches your existing site or create your own.

To select your theme:

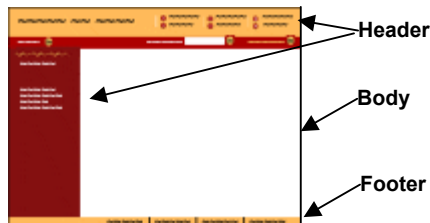
- ◆ Navigate to Store Design.
- ◆ Click the **Change** link next to Theme in the **Design Settings** section. The Theme page opens.
- ◆ Select a category from the **Theme Category** pull-down menu. Browse the theme categories until you find a theme you like.
- ◆ Click the **Select** option for the theme you like.
- ◆ Click the **Save and Publish** button.

-
1. You can create your own theme by selecting the **Create Your Own** theme from the **Favorites** theme category.
 2. The theme is stored in your store version's header and footer templates. Any changes to these templates may alter the store version's theme.
-

Customizing a Theme

ProStores ships with many predesigned themes that you can use. You can browse these themes to see if one matches the general design/colors of your site. If you can't find a perfect match, find one that comes the closest, and then modify the Header and Footer templates so that they match your existing site.

The Header and Footer templates are those portions of your site design that appear on every template. Typically, the Header runs along the top and left portion of the page, and the Footer runs along the bottom, though there are some exceptions.



The Header and Footer Templates

To modify your Header or Footer page templates:

- 1 In Store Administration, navigate to Store Design > Page Template Manager.
- 2 Expand the **Theme** category if necessary by clicking the expand (+) button.
- 3 Click the title of the Header or Footer page template. The template opens in the Page Template Editor.

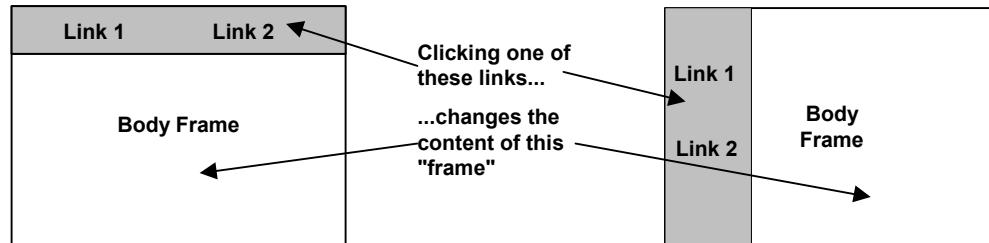
If you're not familiar with HTML, you can use the Basic Editor or the Advanced Editor to edit the template in graphic mode. If you prefer to work directly with the HTML, click the **Source Editor** tab in the Page Template Editor.

- 4 After making any changes to the template, click the **Save** button to save your changes without publishing them, the **Save and Publish** button to publish your changes, or the **Cancel** button to discard your changes.

The Header and Footer templates contain the design elements for your site ie., your store name, store logo and images that appear consistently throughout the site. The Header and Footer templates surround the content area. The content area is dynamic and holds the individual pages for your site.

If Your Existing Site Uses Frames

Some sites are designed using Frames. “Framed” sites divide your pages into separate regions (for example, to contain the navigation in one frame, and the content in another). Typically in a framed site, one region stays the same all the time, but another region will change based the selected links (and usually, these pages will “scroll” independently of each other).



Two examples of framed sites. One frame is shaded, the other is white.

If you created your existing site using frames, you’ll want to modify your store’s theme so that it works with your existing site’s frames. For example, if you have a frameset made of a top banner and body, you’ll want to remove or move the content in your store’s top banner section. If your frameset is made up of a left and right column, you’ll want to modify the theme to move or remove the content on the left side.


In most of the themes, the top and left content of the store is contained in the Header template - modifying this template will allow you to move and remove content so that it fits with your frame design.

Copying Content From Another Web Site

The **Copy Center** tool helps you to copy portions of another web site (such as a site you already maintain) to your storefront. Using **Copy Center**, you can view both your storefront and another web site at the same time, and then “drag and drop” pieces of the web site into your storefront.

This tool is only available if you are using Internet Explorer version 5.5 or higher.

To copy content from another site to your store:

- 1 Navigate to Store Design > Page Template Manager.
- 2 Open the page template that want to copy content into using one of the two methods for opening a page in the Page Template Editor.
 - ◆ Click the title of the page template.
 - ◆ Select the check box next to the page template, and then select **Edit Page Template** from the **More Actions...** pull-down menu.
- 3 Click the **Advanced Editor tab**.
- 4 Click the **Copy Center** button () on the tool bar. The Copy Center opens in a new window.

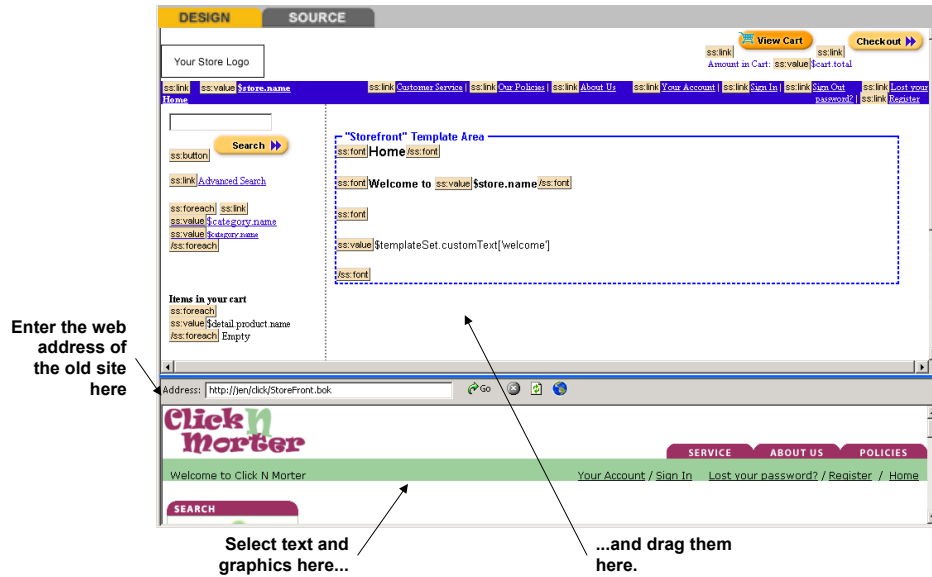
- 5 In the Copy Center window, enter the full HTTP address of the Web page from which you wish to copy content, and click the **Go** button. The Web page appears in the Copy Center window.
- 6 Use your mouse to select the content you wish to copy in the Copy Center, and then drag the content to the appropriate location in the Advanced Editor.

To make content selection easier, right-click inside the Copy Center window and clear the **Browser mode** option.

- 7 In the **Migrate Embedded Image** dialog box, select one of the two URL options:
 - ◆ **Leave the Source URL ...** inserts the image, but continues to point to the original source file (instead of uploading the image to your store).
 - ◆ **Copy the image ...** uploads the image to your store's directory on the server, and then points to that uploaded image.
- 8 (Optional) Select the check box to apply the selection to the remaining images that you copy from this Web page.
- 9 When done, close the Copy Center window.

The following table describes the context menu options available when you click the right mouse button with the cursor pointed inside the Copy Center window.

Option	Description
Copy	Copies selected content to the clipboard for pasting into the page template in the Advanced Editor.
Select Element	Selects the element currently under the cursor.
View > Borders	Toggles display of borders around elements such as images and tables.
View > Markers	Toggles display of markers that graphically represent HTML markup tags such as anchors or image tags.
Find	Opens the browser's Find dialog box for searching the page for specific text.
Browse Mode	Toggles browse mode on and off. <ul style="list-style-type: none"> ◆ If Browse Mode is selected (on), the Copy Center window behaves like a browser window. For example, if you click an image link in browse mode, the linked page opens in the Copy Center window. ◆ If Browse Mode is cleared (off), the Copy Center behaves much like an editor window. For example, if you click an image link with browse mode off, the image becomes selected, and the linked page does not open.



Using the Copy Center Tool

Linking to and from Your Storefront

Once you have finalized the design of your storefront, you'll want to add links from your site to your store, and from your store back to your site.

Create Custom Links to Your Store

The Create Custom Links to Your Store page helps you to create the HTML code you need to add links to your storefront from another web site (or e-mail) or to add an "add products to a cart" form to a static site. You can then copy this link and paste it into your external site.

Do not use this tool to create links for use *within* your storefront.

Custom Links

The fields that appear in the **Custom Links** section depend on your selection in the **Link Type** pull-down menu. For example, if you select **Catalog**, this section displays the following:

- ◆ **Link Type**
- ◆ **Link Text**
- ◆ **Link Target**

If you select **Add Product to Shopping Cart**, this section will also display the following:

- ◆ **Product**
- ◆ **Quantity**

◆ Buy It Now

If you make a selection from a pull-down menu, the code in the **Results** section will automatically update to match your selection. However, if you make a change to a field or a check box, click the **Build Link** button to update the code in the **Results** section.

Field	Description
Link Type	Select the type of link you wish to create from the Type pull-down menu. Certain selections will bring up another list of options from which you can choose. For example, if you wish to create a link to a specific product, select Product from the Type pull-down menu, and then select the specific product to which you wish to create the link from the Product pull-down menu.
Category	The category in which to search for products. Available for Search by Product link only.
Manufacturer	The manufacturer for which to search. Available for Search by Manufacturer link only.
Product	Product search words. Available for Search by Product link only. To search for a specific product, use the Product drop-down list instead.
Keyword	The keyword for which to search. Available for Search by Keyword link only.
Product	Select the product for which you wish to search from the Product pull-down menu.
Quantity	The number of products to add to the cart.
Color	The color of the product to be added to the cart. Available only for products that have color attributes assigned.
Size	The size of the product to be added to the cart. Available only for products that have size attributes assigned.
Buy It Now	Select this check box if the product should be added to the cart and the visitor should go straight to the Checkout page.
Link Text	Enter the text that will appear in the link.
Link Target	From the pull-down menu, select the window in which the page should open: <ul style="list-style-type: none">◆ Open in new window◆ Open in same window

Results

This section displays the results of your link selections, and allows you to test the link to your store.

Field	Description
Link Sample	The Link Sample area provides a preview of how the link will appear, and provides a link that you can click to test how the link will work.

Field	Description
Link Source	This text area shows the text version of the link. This is the text that ProStores will copy to your clipboard if you click the Copy to Clipboard button. The copied text will also include any changes you make in this text area.
Copy to Clipboard Button	If you are using Internet Explorer version 5.5 or higher, click the Copy to Clipboard button to copy the Link Source to the clipboard. If you are using another browser, select the link code and use your browser's Copy tool. You can then paste it into the editor in which you are editing your external page.

Link Manager

You can use the Link Manager page to activate or deactivate specific pages for your store and to redirect certain store page links to point to external Web pages.

Activate or Deactivate Pages

This option enables you to activate or deactivate one or more store pages. Deactivated pages will not appear in your ProStores Web store.

To activate a page:

- ◆ Navigate to Store Design > Page Template Manager > Link Manager.
- ◆ If you have more than one store version, select the version you want to modify from the **Store Version** pull-down menu.
- ◆ In the **Activate or Deactivate Pages** section, select the check box next to each page you want to activate.

You must activate the Customer Service page before you can activate the Frequently Asked Questions, Privacy Policy, and Store Location pages.

- ◆ Click the **Save** button.

To deactivate a page:

- ◆ Navigate to Store Design > Page Template Manager > Link Manager.
- ◆ If you have more than one store version, select the version you want to modify from the **Store Version** pull-down menu.
- ◆ In the **Activate or Deactivate Pages** section, clear the check box next to each page you want to deactivate.

Deactivating the Customer Service page will also deactivate the Frequently Asked Questions, Privacy Policy, and Store Location pages.

- ◆ Click the **Save** button.

Redirect Pages

This option enables you to link specific pages within your store to Web pages that exist at URLs outside of your ProStores Web store. For example, if you have another Web site that has your About Us page, you can direct your store's About Us link to that page. This gives your visitors a seamless transition between your Web site and your store.

This option only applies to active pages. Deactivated pages appear dimmed, and their options are unavailable.

To redirect a ProStores Web store page to another Web page:

- ◆ Navigate to Store Design > Page Template Manager > Link Manager.
- ◆ If you have more than one store version, select the version you want to modify from the **Store Version** pull-down menu.
- ◆ In the **Redirect Pages** section, select the **Use My Other Site's Page** option associated with the page you want to redirect.
- ◆ Enter the full address of the page in the **URL** field (for example, <http://www.mysite.com/about.html>).
- ◆ Repeat the previous two steps for each page you want to redirect.
- ◆ Click the **Save** button.

To stop redirecting a ProStores Web store page to another Web page:

- ◆ Navigate to Store Design > Page Template Manager > Link Manager.
- ◆ If you have more than one store version, select the version you want to modify from the **Store Version** pull-down menu.
- ◆ In the **Redirect Pages** section, select the **Use My ProStores Page** option associated with the page you want to redirect.
- ◆ Repeat the previous two steps for each page you want to redirect.
- ◆ Click the **Save** button.

Chapter 4: Working with ProStores Store Design

ProStores provides several options for working with templates. You can use the built-in template tools in Store Administration, or you can use other HTML editors and import your files.

Prior to customizing the templates, you should have a basic working knowledge of HTML, including tables. You should also have a good idea in mind of how your store will look.

ProStores Store Design behaves differently depending on the type of browser you are using - this guide will provide alternate instructions for the major browser differences. In addition, many of its features require JavaScript.

Edit Page Template

The Edit Page Template page lets you edit a page template in one of up to three editors, select a different page template to edit, and preview your edits.

Depending on your level of technical skill and comfort, you can use one of up to three types of editor for each page template. You can even make a change to a page template in one editor, then switch to another editor to work on other parts of the template.

-
1. ProStores uses a complex set of rules to determine which editors can edit each template. This means that you may see only one editor tab for some pages, and three editor tabs for other pages. The best way to see which editors can edit a template is to simply open the template in the Edit Page Template page.
 2. If you do switch between editors after making a change to a page template, be sure to save the changes when prompted. Otherwise, you will lose your changes when switching between editors.
 3. Custom text entered in the Page Text store design page appears in the Basic Editor and in the Advanced Editor as a **Custom Text** block outlined in a blue dotted line. In both cases you can view or edit the text. If you edit the text in either editor, clicking the **Save** button will also publish the custom text block.
-

Page Template

The **Page Template** pull-down menu gives you quick access to all of your templates from within the Edit Page Template page.

To edit a page template other than the one that is currently open, select the template from the **Page Template** pull-down menu. The template you selected will open in the editor.

Preview

You can preview your page to see how your changes will look before you publish them.

To preview your edits, click the **Preview** link or button (). Your page template will open in a new window.

Editors

ProStores provides three editors to help you modify your templates. Each editor addresses a particular level of template editing complexity and user expertise when working with templates and code.

Please verify that you are making the correct code changes when using the Advanced Editor and Source Editor because incorrect code may make your store unusable. If you set a baseline before editing a template in the Advanced Editor or Source Editor, you will be able to restore your pre-edit store code should a problem arise.

- ◆ **Basic Editor** provides an easy way to edit text and modify templates and settings without having to know store code such as HTML and SSML.
- ◆ **Advanced Editor** provides tools that let you edit your store templates in a graphical environment. Because this editor is the most powerful, it has the highest level of complexity.
- ◆ **Source Editor** lets you edit your page template source code directly as text.

Managing Your Work

Before you begin work on your site, it's important to understand the process of editing and publishing templates.

All templates are contained within store versions. Store versions allow you to customize your store's appearance based on your users' browser settings (among other things), or even allow you to develop different designs that you change based on seasons, holidays, etc. See Store Versions for more information.

If you have created multiple store versions, select the version you wish to work with from the **Store Version** pull-down menu on the Store Design Store Versions page.

The process of editing and publishing involves the following:

- ◆ If you want to back up your existing design, first set a baseline for your templates. See Set a Baseline for more information.
- ◆ The first step in working with templates is to edit a specific template. See Edit Page Template for more information.

- ◆ Once you've edited a template, click the **Save** button to save your work. This saves the changes that you've made, but *does not* make them appear on your store. If you have saved your work on a template (but have not published the template), a **New edits not published** message will appear next to the template in the **Status** column of the Page Template Manager. This allows you to see at a glance which templates have been edited and need to be published.
- ◆ If you do not like the changes you have made since you last published the page template, you can undo them in the Page Template Editor by clicking either the **Undo** button on the toolbar or the **Undo** button at the bottom of the Page Template Editor. This action discards all changes to the page template since the last time you published the template.
- ◆ When you're done working on all your templates, and want to test them, you can preview the store version. See Page Template Manager for more information.
- ◆ Finally, when you're satisfied with your changes and are ready to have them appear to your customers, you can **publish** them to your site. See Publish Pages for more information.
- ◆ This might also be a good time to set a new baseline for your templates. See Set a Baseline for more information.

This table describes what happens with each step in Store Design, the Page Template Manager, and the live storefront.

Action	What happens in Store Design and the Page Template Manager	What happens in the live storefront
Set Baseline	No changes are visible, but the existing design is “backed up” for future reference.	No impact.
Edit	Edits can be viewed as they are made.	No impact.
Save	Changes are saved, and templates are marked with a New edits not published status message.	No impact, but you can “preview” your changes in the storefront with your store data.
Save and Publish	Changes are saved, and the New edits not published status message disappears.	Changes can now be viewed in the live storefront.
Revert to Baseline	All changes since the last baseline are removed.	Live storefront reverts to the last baselined design.

Set a Baseline

The Set a Baseline page lists page templates for which you have chosen to set a baseline.

Click the **Set Baseline** button to save the current versions of the listed templates as a baseline, or click the **Cancel** button to return to the Page Template Manager.

Revert to Baseline

The Revert to Baseline page lists page templates for which you have chosen to restore to the baseline version.

This action will erase any changes made after the last baseline was set. Use with caution, as you cannot undo this action.

Click the **Revert to Baseline** button to return the current versions of the listed templates to baseline versions, or click the **Cancel** button to return to the Page Template Manager.

Revert to Original Version

The Revert to Original Version page lists page templates which you have chosen to restore to their original content.

Reverting to the original version erases all changes you have made. Use with caution, as you cannot undo this action. Also, this action does not apply to page types that you define, because ProStores has no way to determine the state of the original version.

Click the **Revert to Original Version** button to return the current versions of the listed templates to the original versions, or click the **Cancel** button to return to the Page Template Manager.

Publish Pages

The Publish Pages page lists page templates that you have chosen to publish, which makes the pages available to your customers.

To publish the listed page templates, click the **Publish** button.

You may want to strongly consider making a baseline of your templates before and after making significant changes to them. The baseline gives you a set of templates to return to if your changes don't work out quite as planned.

To publish page templates:

- ◆ Navigate to Store Design > Page Template Manager.
- ◆ Select the check box next to each page template you would like to publish.
- ◆ Click the **Publish** button. The Publish Pages page opens.
- ◆ Click the **Publish** button in the Publish Pages page to publish your changes.

Common Tasks

This sections contains topics on common tasks to perform in ProStores Store Design. All these instructions assume you are working in the Advanced Editor unless otherwise noted.

Selecting Content

If you are using Internet Explorer 5.5 or higher, the **Advanced Editor** and **Basic Editor** allow you to select text, tables, images, etc., by using your mouse. The following tips describe how to select different types of content.

Sometimes, you may have difficulty selecting content because a table has been selected (indicated by the handlebars - the small squares that surround the table). To cancel the table selection, you can try clicking somewhere outside of the handlebars, or press the **Tab** key on your keyboard (which moves the selection to the next object on the page), or press the **Esc** key on your keyboard.

Text

To select text (necessary when applying formatting or deleting a block of text), move the cursor to the left of the text you wish to select, press and hold the left mouse button, and “drag” the mouse to the right (and/or down) until you reach the end of the text. Release the mouse button. The text will appear highlighted.



Selected text

Images

You have two options for selecting images.

Normal Selection


To select an image in normal Design Mode view, move the mouse over the image, and then click the left mouse button once. Handlebars will appear around the image. If there is an `<ss:link>` tag surrounding the image, the tag will be highlighted as well.



Selecting images in normal Design Mode view

Marker Selection (Advanced Editor only)

Sometimes you may need to select tiny or “hidden” images. Store Design provides image markers that let you select these images.

To select an image using markers, click  on the toolbar, move the mouse over the “img” marker, and click once. Handlebars will appear around the image. If there is an <ss:link/> tag surrounding the image, the tag will be highlighted as well.



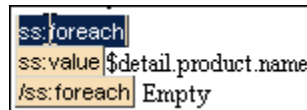
Selecting images using markers

SSML Tags (Advanced Editor only)

To select an SSML tag, move the mouse over the tag icon, and then click the left mouse button once. If the tag is a link or a value, both the tag icon and its text/object expression will appear highlighted. Otherwise, just the opening tag will appear highlighted.



Selecting ss:link and ss:value Tags

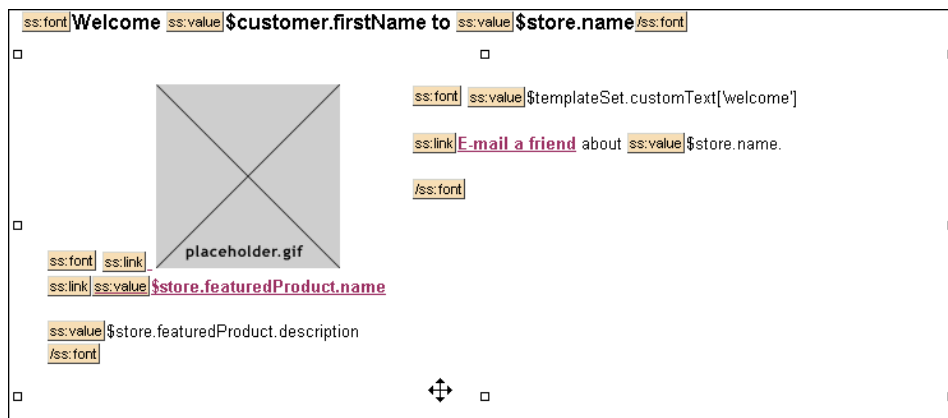


Selecting all other tags


Tables (Advanced Editor only)

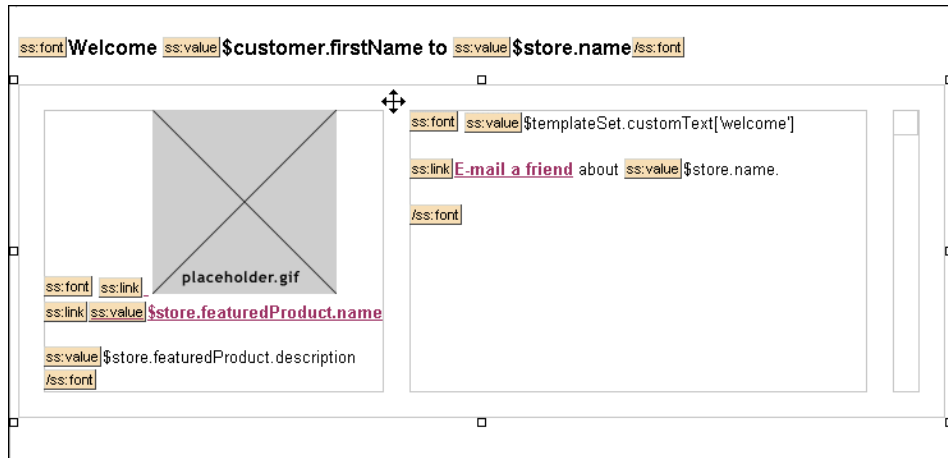
There are several tools to assist you in selecting tables.

If you know where your table borders are, move the mouse over any border and click the left mouse button once. The table will be selected, as indicated by the handlebars.





Selecting tables

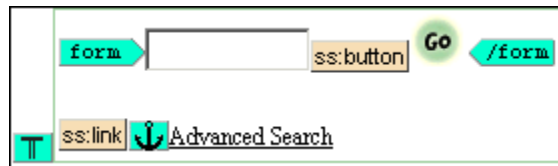
If you do not know where your borders are, click  on the toolbar - the table borders will appear. Move the mouse over any border, and then click the left mouse button once. The table will be selected, as indicated by the handlebars.



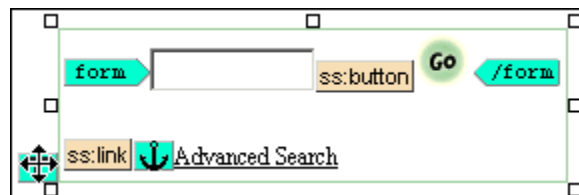
Selecting tables using borders

You can also select tables by displaying the markers.

To display the markers, click , move the mouse over the  marker, and then click the left mouse button once. The table will be selected, as indicated by the handlebars.




Selecting tables using markers. First find the marker...



...then select it


Formatting

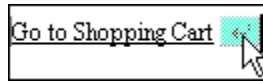
To select a block of content that is formatted (for example, bold or italic text), click , move the mouse over the appropriate marker, and then click the left mouse button once. All text within that formatting tag will appear selected.



Selecting formatting

Special Characters (Advanced Editor only)

To select special characters such as line breaks, paragraph marks, or comments, click , move the mouse over the appropriate marker, and then click the left mouse button once. The marker (and therefore the special character) will appear highlighted.



Selecting special characters

Adding Text

Store Design provides two ways to select text, depending on which browser you are using to edit your page template.

If you are using Internet Explorer 5.5 or higher

To add text to a template:

- 1 Place the cursor in the location in which you want to add text. The cursor appears as a flashing vertical bar (|).
- 2 Type the text you wish to add.

For all other browsers

To add text to a template:


- 1 Click on any existing content. A menu appears.
- 2 Select either **Insert Text Before...** or **Insert Text After...** to specify the exact location of the text. The Edit Text page appears.
- 3 Enter the text you wish to add.
- 4 If necessary, specify any formatting options.
- 5 Click **Save** when done. The text appears at the specified location.

Adding Links

Store Design provides two ways to add hyperlinks, depending on which browser you are using to edit your page template.

If you are using Internet Explorer 5.5 or higher

To add a link to a template:

- 1 Place the cursor in the location where you want the link to appear. The cursor appears as a flashing vertical bar (|).
- 2 Click the **Insert/Edit hyperlink** button () on the toolbar. The **Add/Edit Link** window appears.
- 3 At this point, you have three options:
 - ◆ **Pre-Built Links:** The first portion of the **Link Type** list contains frequently used, pre-built links. To use one of these links, select it from the list. If the link requires additional information, it will appear within square brackets ([...]) in the second field. Enter the necessary information.
 - ◆ **<ss:link>:** Select an `<ss:link/>` tag from the **Link Type** list. If you select this option, you'll then need to enter an object expression in the second field. For help on selecting an object expression, click **Help**.
 - ◆ **Legacy Tags:** Select a legacy tag from the **Link Type** list (legacy tags start with "<:"). However, these tags are intended only for use by stores using older versions of ProStores.
- 4 If necessary, enter the text you wish to have appear hyperlinked in the **Link Value** field.
- 5 If you wish to have this link open in a new window, select that option from the **Target** pull-down menu.
- 6 Click the **OK** button. The link will appear in the location you specified.

For all other browsers

To add a link to a template:

- 1 Click on any existing content. A menu appears.
- 2 Select either **Insert Link Before...** or **Insert Link After...** to specify the exact location of the link. The Edit Link page appears.
- 3 At this point, you have three options:
 - ◆ **Pre-Built Links:** The first portion of the **Link Type** list contains frequently used, pre-built links. To use one of these links, select it from the list. If the link requires additional information, it will appear within square brackets ([...]) in the second field. Enter the necessary information.
 - ◆ **<ss:link>:** Select an `<ss:link/>` tag from the **Link Type** list. If you select this option, you'll then need to enter an object expression in the second field. For help on selecting an object expression, click the **Help**.
 - ◆ **Legacy Tags:** Select a legacy tag from the **Link Type** list (legacy tags start with "<:"). However, these tags are intended only for use by stores using older versions of ProStores.


- 4 If necessary, enter the text you wish to have appear hyperlinked in the **Link Value** field.
- 5 If you wish to have this link open in a new window, select that option from the **Target** pull-down menu.
- 6 If necessary, select any formatting options.
- 7 Click the **Save** button. The link will appear in the location you specified.

Adding Images

Store Design provides two ways to add images, depending on which browser you are using to edit your page template.

If you are using Internet Explorer 5.5 or higher

To add an image to a template:

- 1 Place the cursor in the location in which you want the image to appear. The cursor appears as a flashing vertical bar (|).
- 2 Click the **Insert/Edit Image** button () on the toolbar. The **Image** window appears.
- 3 At this point, you have two options:
 - ♦ **Specific Images:** you can point to a specific image that you've already uploaded to your store. This is recommended primarily when adding store design images. To do so, select the **Store Image** option, and select an image from the list.
 - ♦ **Dynamic Images:** if you wish to insert a "dynamic" image (for example, you want to display a product image), select the **URL/Source/Shortcut** option. Then enter the appropriate URL, object expression or shortcut. For help on selecting an object expression or shortcut, click the **Help** button.
- 4 Enter text to describe the image in the **Alternate Text** field. This can be very helpful in making your site accessible to users with disabilities.
- 5 If necessary, select any other formatting options.
- 6 Click the **OK** button. The image will appear in the location you specified.

For all other browsers

To add an image to a template:

- 1 Click on any existing content. A menu appears.
- 2 Select either **Insert Image Before...** or **Insert Image After...** to specify the exact location of the image. The Edit Image page appears.
- 3 At this point, you have two options:
 - ♦ **Specific Images:** you can point to a specific image that you've already uploaded to your store. This is recommended primarily when adding store

design images. To do so, select the **Store Image** option, and select an image from the list.

- ♦ **Dynamic Images:** if you wish to insert a “dynamic” image (for example, you want to display a product image), select the **URL/Source/Shortcut** option. Then enter the appropriate URL, object expression or shortcut. For help on selecting an object expression or shortcut, click the **Help** button.
- 4 Enter text to describe the image in the **Alternate Text** field. This can be very helpful in making your site accessible to disabled users.
 - 5 If necessary, select any other formatting options.
 - 6 Click the **Save** button. The image will appear in the location you specified.


Display Store Data

Store Design provides two ways to display store data, depending on which browser you are using to edit your page template.

Store data includes store address and contact information, system information, customer information, etc.

If you are using Internet Explorer 5.5 or higher

To display store data:

- 1 Place the cursor in the location in which you want to display the data. The cursor appears as a flashing vertical bar (|).
- 2 Click the **Insert/Edit Tag** button () on the toolbar. The **Insert/Edit Tag** window appears.
- 3 At this point, you have the following options:
 - ♦ **Frequently Used Data:** select the data you wish to display from the Frequently Used Tags portion of the **Tags** list.
 - ♦ **<ss:value/> Tag:** if the data you wish to display isn't in this list, then select ProStores Value Tags from the **Tag Type** list, and then select the `<ss:value source=" " />` tag. Enter the object expression that represents the data you wish to display (for example, \$store.name). For help on selecting an object expression, click the **Help** button.
- 4 Click the **OK** button. The ss:value tag icon will appear in the location you specified, with the data source appearing to its right.

For all other browsers

To display store data:

- 1 Click on any existing content. A menu appears.

- 2 Select either **Insert Tag Before...** or **Insert Tag After...** to specify the exact location of the link. The Edit Tags page appears.
- 3 At this point, you have the following options:
 - ◆ Select the data you wish to display from the Frequently Used Tags **Tags** list.
 - ◆ If the data you wish to display isn't in this list, then select ProStores Value Tags from the **Tag Type** list, and then select the `<ss:value source="" />` tag. Enter the object expression that represents the data you wish to display (for example, `$store.name`). For help on selecting an object expression, click the **Help** button.
- 4 Click the **OK** button. The **ss:value** tag icon will appear in the location you specified, with the data source appearing to its right.


Building Lists

Lists allow you to display the same information for a group of items. For example, you may want to display a list of the product name and price for all items returned in a search. Or, you may want to display a list the date and status for each order a customer has made in your store.

Store Design provides two ways to build lists, depending on which browser you are using to edit your page template.

If you are using Internet Explorer 5.5 or higher

To add a list to a template:

- 1 Place the cursor in the location where you want to add the list. The cursor appears as a flashing vertical bar (|).
- 2 Click the **Insert/Edit Tag** button () on the toolbar. The **Insert/Edit Tag** window appears.
- 3 From the **Tag Type** list, select **ProStores Other Tags**.
- 4 From the **Tags** list, select the `<ss:foreach item="" within=""></ss:foreach>`.
- 5 Enter the name you will use for the items in the list in the *item* attribute, then enter the object expression that represents the source of the list in the *within* attribute. For help on selecting an object expression, click the **Help** button.
- 6 Click the **OK** button. The **ss:foreach** tag icons will appear in the location you specified.
- 7 Add the items you wish to have appear in the list:
 - ◆ Place the cursor between the **ss:foreach** tag icons. One way to do this is to press the arrow keys on your keyboard until the cursor is between the icons. If you use your mouse to place your cursor between the icons, you may end up *editing* the `<ss:foreach>` tag you just entered instead of entering a new

tag (you'll know this because you'll see the tag in the Markup field). If this occurs, click the **Cancel** button, and move your cursor using the arrow keys.

- ◆ Add text, tags, and images as necessary by clicking the appropriate button on the toolbar.

For all other browsers

To add a list to a template:

- 1 Click on any existing content. A menu appears.
- 2 Select either **Insert Tag Before...** or **Insert Tag After...** to specify the exact location of the link. The Edit Tags page appears.
- 3 From the **Tag Type** list, select **ProStores Other Tags**.
- 4 From the **Tags** list, select the `<ss:foreach item="" within=""></ss:foreach>`.
- 5 Enter the name you will use for the items in the list in the *item* attribute, then enter the object expression that represents the source of the list in the *within* attribute. For help on selecting an object expression, click **Help**.
- 6 Click **Save**. The **ss:foreach** tag icons will appear in the location you specified.
- 7 Add the items you wish to have appear in the list:
 - ◆ Click on the first **ss:foreach** tag icon. A menu appears.
 - ◆ Add text, tags and images as necessary by clicking the appropriate button on the toolbar.


Using Conditions

Conditions allow you to control the display of information based on customer properties, store type, system information, etc. For example, you might display a special message to a returning customer, or you could display a certain image on a specified holiday.

Store Design provides two ways to display content based on a condition, depending on which browser you are using to edit your page template.

If you are using Internet Explorer 5.5 or higher

To display content based on a condition:

- 1 Place the cursor in the location in which you want to add the content. The cursor appears as a flashing vertical bar (|).
- 2 Click the **Insert/Edit Tag** button () on the toolbar. The **Insert/Edit Tag** window appears.
- 3 From the **Tag Type** list, select **ProStores Other Tags**.

- 4 To control the display of information based on whether or not a single condition is met, select the `<ss:if test=""></ss:if>` tag:
 - ◆ Enter the condition that must be met within the *test* attribute. For help on creating test conditions, click the **Help** button.
 - ◆ Enter the content that is to appear if the condition is met between the `<ss:if test=""></ss:if>` tags. If you want to display content if the condition *is not met*, type `<ss:else/>`, and then type the alternate content.
- 5 To control the display of information based on whether one of several conditions is met, select the `<ss:choose></ss:choose>` tag. Now you need to add your test conditions:
 - ◆ Place the cursor between the `<ss:choose></ss:choose>` tags.
 - ◆ Type `<ss:when test=""></ss:when>`.
 - ◆ Enter the condition that must be met within the **test** attribute. For help on creating test conditions, click the **Help** button.
 - ◆ Enter the content that is to appear if the condition is met between the `<ss:when test=""></ss:when>` tags.
 - ◆ Continue adding `<ss:when/>` tags until you have entered all test conditions.
 - ◆ If you want to display content when *none* of the conditions are met, add an `<ss:otherwise>` tag (using the same process).
- 6 Click the **OK** button. The conditional content will appear (the `ss:if`, `ss:choose` and `ss:when` tag icons do not appear in Store Design).

For all other browsers





To display content based on a condition:

- 1 Click on any existing content. A menu appears.
- 2 Select either **Insert Tag Before...** or **Insert Tag After...** to specify the exact location of the link. The Edit Tags page appears.
- 3 From the **Tag Type** list, select **ProStores Other Tags**.
- 4 To control the display of information based on whether a single condition is met, select the `<ss:if test=""></ss:if>` tag.
 - ◆ Enter the condition that must be met within the *test* attribute. For help on creating test conditions, click the **Help** button.
 - ◆ Enter the content that is to appear if the condition is met between the `<ss:if test=""></ss:if>` tags. If you want to display content if the condition *is not met*, type `<ss:else/>` and then type the alternate content.
- 5 To control the display of information based on whether one of several conditions is met, select the `<ss:choose></ss:choose>` tag. Now you need to add your test conditions.

- ◆ Place the cursor between the `<ss:choose></ss:choose>` tags.
 - ◆ Type `<ss:when test=""></ss:when>`.
 - ◆ Enter the condition that must be met within the *test* attribute. For help on creating test conditions, click the **Help** button.
 - ◆ Enter the content that is to appear if the condition is met between the `<ss:when test=""></ss:when>` tags.
 - ◆ Continue adding `<ss:when/>` tags until you have entered all test conditions.
 - ◆ If you want to display content if *none* of the conditions are met, add an `<ss:otherwise>` tag (using the same process).
- 6 Click the **OK** button. The conditional content appears (the `ss:if`, `ss:choose` and `ss:when` tag icons do not appear in Store Design).

Saving and Undoing Your Work

The following table details your save, undo and delete options.

Function	Button	Description
Save		Saves your changes without making them “live.” (Internet Explorer 5.5 and higher only)
Save and Publish		Saves your changes <i>and</i> makes them live.
Undo		Undoes all changes since the last time you <i>saved</i> the template.
Delete		Deletes the current template. Only appears on custom templates you have added.

Advanced Tasks

This sections contains topics on more advanced tasks you can perform in ProStores Store Design.

Validate Page Template

You can use the **Validate Page Templates** button on the Page Template Manager to test the validity of the SSML code in your templates. If one or more templates fails the validation test, an “Invalid Code” message appears in the Status column for each template that failed. If all templates pass the validation test, a “success” message appears.

Any templates opened in Page Template Editor are automatically validated.

The page template validation tests SSML only; it does not test for valid HTML.

Validate Page Templates

To validate a template:

- ◆ Navigate to Store Design > Page Template Manager.
- ◆ If you have more than one store version, select the version you want to validate from the **Store Version** pull-down menu.
- ◆ Click the **Validate Page Templates** button.
- ◆ Check the **Status** column for any “Invalid Code” messages.

Import and Replace Page Template

You can use the Import and Replace Page Template page to select a new template that will overwrite the specified template. One **New ...** field will appear for each template selected in the Page Template Manager.

The **New ...** field label lists the template that the new template will replace. For example, if you selected a template named “Product Detail” to replace, the field label will appear as **New Product Detail**. This label design helps you quickly determine which template you are replacing, which is especially helpful when you have a list of templates for replacement.

New ...

For each template listed, click the **Browse...** button to select the template that will overwrite the associated template. After making your selections, click the **Import and Replace** button to complete the action, or click the **Cancel** button to discard your selections and return to the Page Template Manager.

Export Page Template

You can use the Export Page Template page to select the operating system in which you will use the exported files.

After you click the **Export** button, the Export Page Template page will display a link to a ZIP file that contains all page templates selected for export. You must save this file, then extract the page templates using a standard extraction program (such as WinZip™), before you can use the page template files.

Operating System

Select the operating system in which you will use the exported files from the **Operating System** pull-down menu.

Operating systems have their own way of handling formatting, such as line endings in text files. If you see mysterious characters when you open exported files, you may want to try exporting again with a different operating system selected.

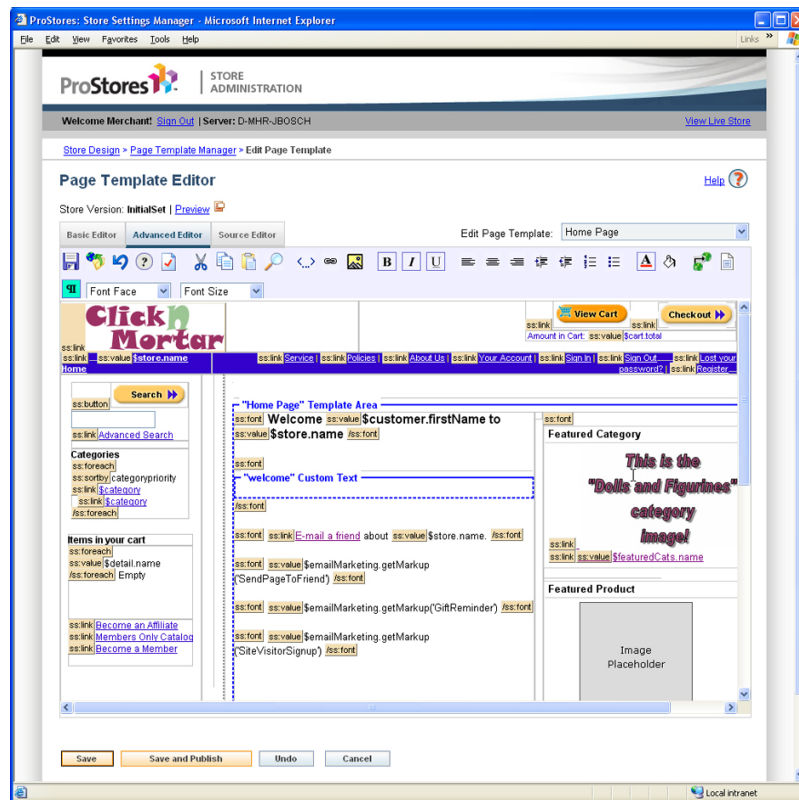
Using Template Edit Modes

ProStores Store Design offers two “modes” in which you can edit your templates, design mode and source mode. Design mode presents the page template and editing tools in a graphical format, while source mode presents the page template as pure text. The **Basic Editor** and **Advanced Editor** work in design mode; the **Source Editor** works in source mode.

Design vs. Source Mode

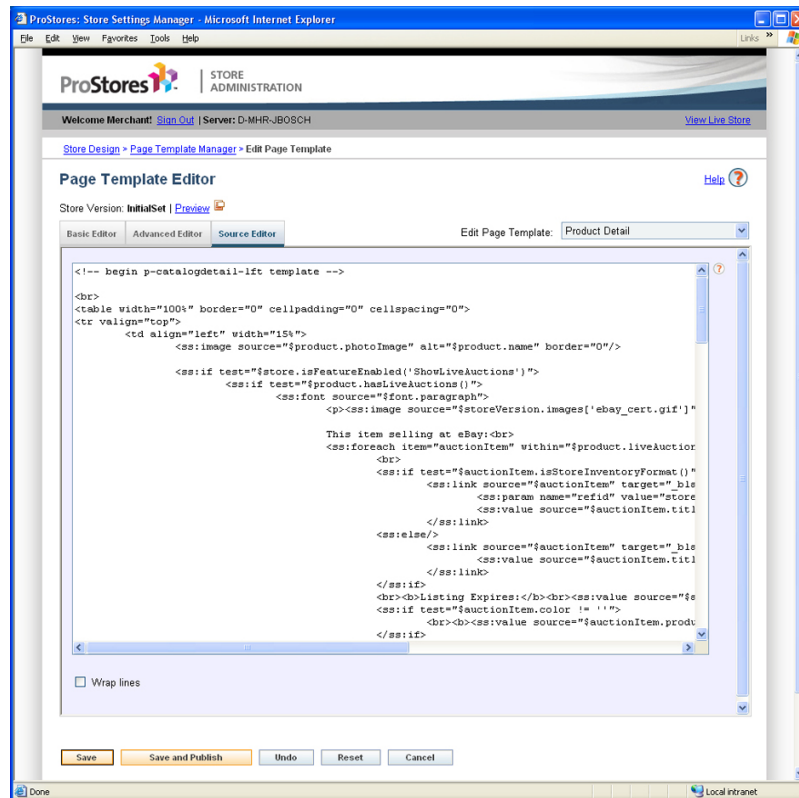
Most page templates will open in design mode in the **Advanced Editor**. This mode gives you a What-You-See-Is-What-You-Get (“WYSIWIG”, pronounced “Wizzy Wig”) view of your template, including the Header and Footer template. Two editors use design mode:

- ◆ **Basic Editor**
presents a simplified view of the page template. You can edit the template without having to know anything about HTML or SSML. This is the easiest editor to use, but is not as powerful as the **Advanced Editor**.
- ◆ **Advanced Editor**
also presents the page template in a graphical view. However, because you have control over adding and editing SSML tags, you have greater control over your page template contents and layout.



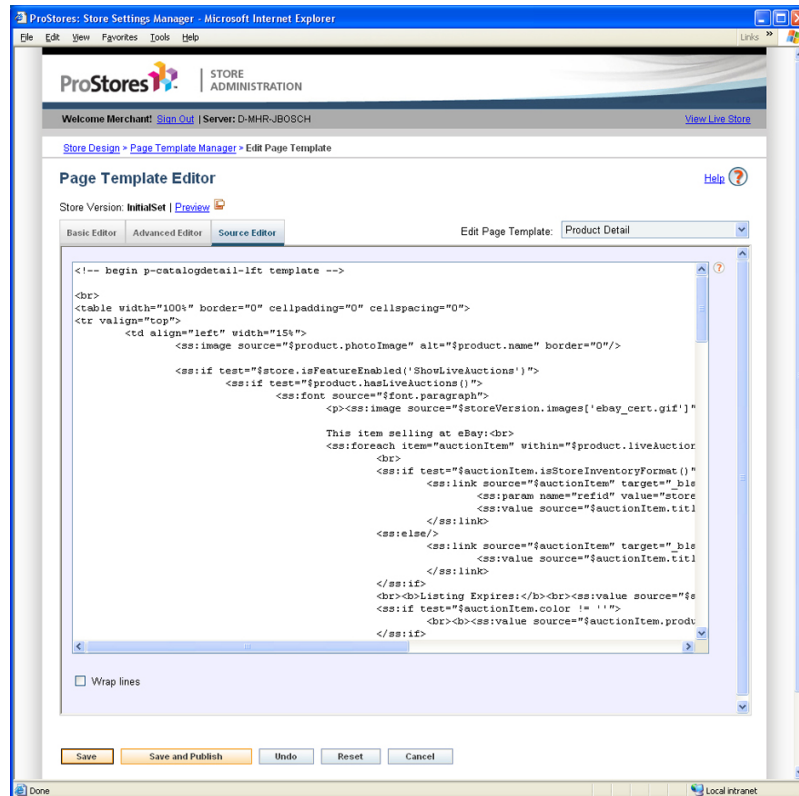
Design Mode

You can edit any page template in source mode using the **Source Editor**. The **Source Editor** displays the source code of the template as plain text, but does not include the header or footer template code. Use the **Source Editor** with care!



Source Mode

Other templates - like templates used to generate e-mails - only open in the **Source Editor**.



Template Source Editor

See Also

Using the Advanced Editor

Using the Template Source Editor

Browser Differences

While you can use ProStores Store Design in any browser version that supports HTML 4.0 and Cascading Style Sheets (such as Microsoft Internet Explorer 4.0, Netscape 6.2, FireFox 1.5, or any higher version of these browsers), it is optimized for use with Internet Explorer 5.5 or higher.

- ◆ If you use Internet Explorer 5.5 or higher, you can do all your editing and formatting using the Template Toolbar.
- ◆ If you use Netscape 6.2 or higher, or FireFox 1.5 or higher, you will be able to add or insert elements using a “pop-up” menu, and then do all your editing and formatting on the resulting page.
- ◆ If you use another browser, you will be able to add or insert elements by making a selection from a form, and then do all your editing and formatting in the resulting page.

Basic Editor

The Basic Editor, like the Advanced Editor, gives you a WYSIWYG (“what you see is what you get”) view of your templates, with the Header and Footer templates active for editing as well. In the Basic Editor, your page appears almost exactly as it will appear to your customers, with the exception of some ProStores tags (which appear as tags and not as their resolved values).

For the most part, the Basic Editor works just like a word processor, complete with the familiar tool bar and context (right-click) menu. However, there are some important differences, so please consider the following points when editing your templates:

- ◆ Although Internet Explorer versions 5.5 and higher allow you to edit text directly in the Basic Editor, do *not* type any source code (such as HTML or ProStores tags) directly into the page template in the Basic Editor. Instead, use the Advanced Editor to enter or edit these tags. If you prefer to work with source code, you can switch to the Source Editor.
- ◆ When saving a template in the Basic Editor, ProStores may reformat your source code (e.g., white space may be removed, element names will be capitalized, etc). This will not change the content of your code, or affect the look of your template in any way. However, if you prefer to retain your source code formatting, you may want to work exclusively in the Source Editor, or export your template, edit it in an editor program of your choice, and then import your finished template back into ProStores.
- ◆ Templates open in the Basic Editor complete **with** the Header and Footer template as well, allowing you to edit all three templates. Changes made to the Header or Footer area when working on another template *will affect the look of your entire store*. If you make changes to these areas, the Header and Footer templates will display the “in progress” symbol. In addition, when you save (or save and publish) a template from the Basic Editor, the Header and Footer templates will be saved (or saved and published) as well.
- ◆ If you are using Internet Explorer version 5.5 or higher, the first time you use the Basic Editor, you may be prompted to “Run Store Design”. This installs the control required for some of the Basic Editor’s features. Click **Yes**, and the control will be installed.
- ◆ Custom text entered in the Page Text store design page appears in the Basic Editor as a **Custom Text** block, outlined in a blue dotted line, in which you can view or edit the text. If you edit the text in the **Custom Text** block, clicking the **Save** button will also publish the changes made inside the **Custom Text** block.

Using the Advanced Editor

The Advanced Editor, like the Basic Editor, gives you a WYSIWYG (“what you see is what you get”) view of your templates, with the Header and Footer templates active for editing as well. In the Advanced Editor, your page appears almost exactly as it will appear to your

customers, with the exception of the ProStores tags (which appear as tags and not as their resolved values).

For the most part, the Advanced Editor works just like a word processor, complete with the familiar tool bar and context (right-click) menu. However, there are some important differences, so please consider the following points when editing your templates:

- ◆ Although Internet Explorer versions 5.5 and higher allow you to edit text directly in the Advanced Editor, do *not* type any source code (such as HTML or ProStores tags) directly into the page template in the Advanced Editor. Instead, use the Tag Element to enter or edit these tags. If you prefer to work with source code, you can switch to the Source Editor.
- ◆ When saving a template in the Advanced Editor, ProStores may reformat your source code (e.g., white space may be removed, element names will be capitalized, etc). This will not change the content of your code, or affect the look of your template in any way. However, if you prefer to retain your source code formatting, you may want to work exclusively in the Source Editor, or export your template, edit it in an editor program of your choice, and then import your finished template back into ProStores.
- ◆ Templates open in the Advanced Editor complete **with** the Header and Footer template as well, allowing you to edit all three templates. Changes made to the Header or Footer area when working on another template *will affect the look of your entire store*. If you make changes to these areas, the Header and Footer templates will display the “in progress” symbol. In addition, when you save (or save and publish) a template from the Advanced Editor, the Header and Footer templates will be saved (or saved and published) as well.
- ◆ If you are using Internet Explorer version 5.5 or higher, the first time you use the Advanced Editor, you may be prompted to “Run Store Design”. This installs the control required for some of the Advanced Editor’s features. Click **Yes**, and the control will be installed.
- ◆ Custom text entered in the Page Text store design page appears in the Advanced Editor as a **Custom Text** block, outlined in a blue dotted line, in which you can view or edit the text. If you edit the text in the **Custom Text** block, clicking the **Save** button will also publish the changes made inside the **Custom Text** block.

Template Toolbar

The Template Toolbar appears when editing a template in Design Mode, and provides quick access to key template editing tasks, depending on the browser version you are using.

Internet Explorer 5.5 and Higher

- ◆ Save functions (save, publish, undo since publish). See *Saving and Undoing Your Work* (page 32) for more information.
- ◆ Delete function (only available for custom templates).
- ◆ Help on the current template.
- ◆ Formatting Functions (bold, font color, alignment, lists etc).
- ◆ Editing Functions (copy, paste, etc.).

- ◆ Element Functions (insert or edit element).
- ◆ Visual Aids (show gridlines, show markers).

Other Browsers

If you are working in a browser other than Internet Explorer 5.5, the only functions that will appear on the toolbar are Save and Publish, Undo, Delete (if applicable) and the Properties Element button.

The Context Menu

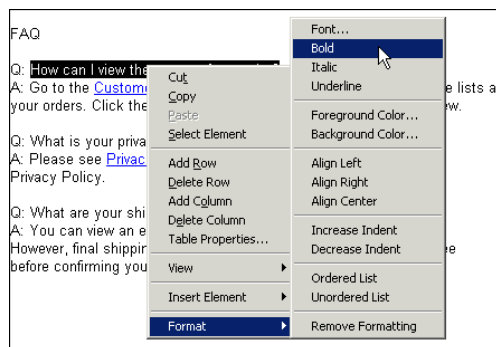
If you are using Internet Explorer version 5.5 and higher, you can use the following tools and commands to help make editing easier.

The Context Menu

If you are using a mouse with two or more buttons, ProStores offers a “context” menu to provide quick access to common functions. Some of these functions include basic editing (cut, copy, paste), formatting (bold, italics), selecting a visual aid, or inserting and adding HTML code.

To access the context menu:

- ◆ Place the cursor in the location where you want to add or edit content.
- ◆ Press the *right* mouse button. The context menu appears.
- ◆ Move the cursor to select (highlight) an action.
- ◆ Press the *left* mouse button.



Using the Context Menu

Keyboard Shortcuts

Store Design provides several keyboard shortcuts to simplify editing tasks in the Basic Editor and in the Advanced Editor.


Action	Keys
Move the cursor	Arrow keys

Copy	CTRL + C
Paste	CTRL + V
Cut	CTRL + X
Undo	CTRL + Z
Redo	CTRL + Y
Bold	CTRL + B
Italic	CTRL + I
Underline	CTRL + U


Visual Aids (Advanced Editor only)

ProStores offers several visual aid tools to help you understand the design and layout of your page.

Show Borders displays a light gridline around all tables and images. This option is enabled by default in Store Design. Borders can help you understand the structure of your page and place elements in the correct location.

To hide borders, click the **Show/hide borders** button () on the tool bar. Click the button again to show the borders.

Show Markers provides visual descriptions of the HTML code on your page, such as line breaks, comment codes, and anchor tags (hyperlinks).

To enable markers, click the **Show/hide markers** button () on the tool bar. Click the button again to hide markers.

In addition to borders and markers, most SSML tags appear as icons. Exceptions include images and some advanced tags.


Some SSML tag icons may not appear in the location in which you expect them. For example, some `<ss:foreach/>` tags may appear outside of the table in which they are located. This is due to the tag's location in the source code, and does not affect the functionality of the tag.

Catalog List Sorting and Grouping

This tool only applies to templates using legacy ProStores tags - specifically, the `<:groupby>` and `<:sortorder>` tags.





ProStores Store Design Advanced Editor simplifies the process of creating sort order and group headings on the Catalog List templates. “Sorting” refers to the order in which products are displayed - you can sort on up to three levels (for example, by category, then by subcategory, then finally product). Group headings allow you to apply special formatting or titles that appear for each unique group (defined by your sort criteria). For example, you may choose to display the category name and image at the beginning of each group of categories.

To set your sort order and group headings on a Catalog List template:

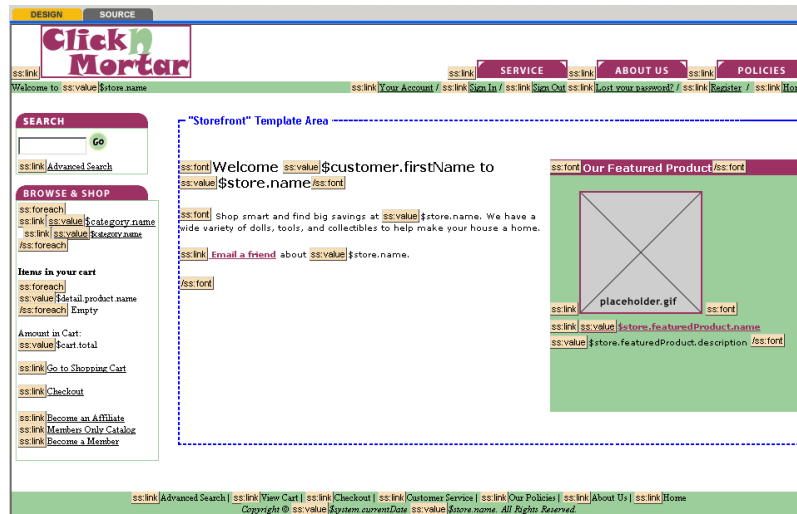
- 1 On a Catalog List template, click . The Edit Template Sort Order window appears.
- 2 Using the lists, select the primary, secondary and tertiary sort orders.
- 3 If you wish to display headings or formatting for each sort grouping, enter the code in the **Heading** box.
- 4 Click **OK** or **Save** when done. You are returned to the Catalog List template.

Elements

When working in the **Advanced Editor** or **Basic Editor**, ProStores Store Design can automate the process of adding tags, images and links - otherwise known as “elements.” There are several types of elements (the button images reflect the Template Toolbar button that accesses the element in Internet Explorer 5.5 or higher):

Element	Button	Description
Text	n/a	Inserts and edits text. See Text Elements for more information.
Tag		Advanced Editor only: Inserts or edits an HTML or ProStores tag. See Tag Elements for more information.
Image		Inserts or updates an image. See Image Elements for more information.
Link		Inserts or edits an HTML, MAILTO or ProStores tag link. See Link Elements for more information.
Properties		Advanced Editor only: Sets general page properties. See Changing Template Properties for more information.

Tags and links are represented by SSML tag icons. Images are either represented by the actual image, or by a placeholder image.



Identifying Element Types Using Tag Icons

Using the Template Source Editor

In addition to WYSIWYG template editing using the Basic Editor or the Advanced Editor, you can use the Source Editor to edit page template source code directly. Working with the Source Editor is much like working in a basic text editor. Source code appears as plain text, but does not include the header and footer template code (unless you're editing the header or footer template, of course).

Some templates are used within other templates. Many of these “child templates”, such as the Low Inventory - Email template, only open in the Source Editor. In addition to not having the header or footer source code, the Source Editor also does not use a tool bar. However, the **Save**, **Save and Publish**, **Undo**, **Reset**, and **Cancel** buttons work the same as they do for the other editors. In addition, **Reset** undoes all changes since the last save.

When working with template source code, it is important that you avoid using the ASCII equivalent of the ampersand symbol (&). If you must display an ampersand on your page, use '&', *not* '&'.

Wrap Lines

The Source Editor supports word wrap, a feature that continues the display of a long line on the next line, instead of allowing the code to run past the right edge of the editor window. Enabling word wrap removes the need of having to use the horizontal scroll bar to view long lines.

Select the **Wrap lines** check box to cause the Source Editor to wrap long lines, or clear the check box to view the lines without wrapping.

Chapter 5: Using Store Versions

A store version (formerly called a template set) is the set of page templates that make up the look and feel of your store. You can create multiple store versions that can customize a user's experience, such as having a store version for a particular holiday, or that are for draft development.

This chapter provides instructions on using store versions.

Understanding Store Versions

With ProStores you can create multiple designs of your store, without having to duplicate your store, customer or product information. You can think of a store version as a new storefront rather than as a new store. With the new storefront, the inventory and other store records remain unchanged.

Some examples of reasons why you might create multiple designs are:

- ◆ You want to change the design of your store based on the season or time of year.
- ◆ You have a large number of customers who use older browsers, and you want to customize the design of your store for them.
- ◆ You have customers in different countries, and you want to create a version of your store in different languages.
- ◆ You want to completely redesign your store, but want to do so without touching your current store's design.

ProStores does this by allowing you to create multiple store versions. Each store version contains the complete set of templates that make up a store design. When you first purchase a ProStores Web store, your store contains one store version, which you can then design using the ProStores Store Design. Most likely, this is all you'll need in order to develop a functional and creative store design.

However, if you want to create an additional design, as described in the examples above, you can do so by adding another store version. You can create a blank design or create a new version based on your existing design using the Copy Store Version action.

Once you've created your additional design, you can then specify when you want to display that design to your customers. If you want to replace your old design with your new one, you would edit the store status to make the original store version inactive, and then make the new store version the default. If you want to have both store versions active, but display one to one type of browser user or language, then you would modify the rules for that store version to only display when a certain browser setting is in effect. See *Edit Rules* (page 46) for more information on editing browser rules, or *Edit Language/Locale* (page 46) for editing language rules.

Store Versions

You can use the Store Versions page to view the status and locale settings for all of your store versions, add new store versions, edit a store version status, and a whole lot more.

Add Store Version

Offering multiple store versions can help you to fine-tune your customer's experience. For example, you can have one version for customers who use Internet Explorer to visit your store, and another for customers who use Mozilla Firefox, then define rules that tell ProStores to display the correct version according to browser type. You can also use store versions to create special-occasion stores, such as a Christmas store or a Halloween store, or to create stores for visitors from different countries.

When you create a new store version, the new version uses the same underlying store, customer, and Product information. You can think of it as a new, customizable storefront instead of as a new store.

To add a store version:

- ◆ Navigate to Store Design > Store Versions.
- ◆ Click the **Add Store Version** button. The Add Store Versions page opens.
- ◆ Enter a name for the new store version.

The store version name cannot contain spaces or the following special characters: (“ , ‘ \ * ? ` | : + & > < ; . “).

- ◆ (Optional) Select a different local from the **Language/Local** pull-down menu.
- ◆ Click the **Save** button.

After you add a store version, you must edit the version status to make it available to your customers.

See Also

Edit Rules

Add Store Version

Edit Status

Edit Status

You can make each store version individually available or unavailable to your customers, and you can specify which store version appears as the default version. The default version will appear to your customers when store version rules don't apply or have not been set.

Click the **Edit Status** button to modify the status for one or more of your stores.

See Also
Edit Status

Edit Rules

Store version rules allow you to tell ProStores which store version to display to a customer based on information that ProStores reads from the customer's browser. For example, you can create a store version for customers who use an older browser, and then create a rule that tells ProStores to display that store version when the customer's browser matches the one specified in the rule.

Select the check box next to one or more store versions, and then click the **Edit Rules** button to modify the rules for each selected store version. You can then add, modify, or delete the custom rules for the store version.

See Also
Edit Rules

More Actions...

The **More Actions...** pull-down menu contains additional actions that apply to selected store versions. Information about each action appears below.

Edit Language/Locale

You can use this setting to tell ProStores to display a particular store version when a customer's Web browser is configured for a particular language.

Select the check box next to one or more store versions, and then select **Edit Language/Locale** from the pull-down menu. You can then select a new language/local.

See Also
Edit Rules

Copy Store Version

You can use this action to make a copy of selected store versions.

Select the check box next to one or more store versions, and then select **Copy Store Version** from the pull-down menu. You can then enter a name and set the language/locale for the new versions.

See Also

Copy Store Version

Delete Store Version

You can use this action to delete selected store versions.

Select the check box next to one or more store versions, and then select **Delete Store Version** from the pull-down menu. You can then confirm or cancel your decision to delete a store version.

See Also

Delete Store Version

Edit Language/Locale

Use the Edit Language/Locale page to specify the language and locale for the selected store versions.

You can use this setting to tell ProStores to display a particular store version when a customer's Web browser is configured for a particular language.

If you have selected more than one store version in the Store Versions page, each will appear in its own section on this page.

Language/Locale

Select the appropriate language/locale combination from the **Language Locale** pull-down menu.

After making your selections, click the **Save** button to save your changes, or click the **Cancel** button to return to the Store Versions page.

Edit Rules

The Edit Rules page allows you to tell ProStores which store version to display to customers based on information their browser sends to the application. For example, if customer access your store from an older browser, you can create a custom store version for those customers,

and then apply a rule that tells ProStores to display that version when your store is visited by customers using a browser that matches the rule. If the browser information does not match the rules of any store version, customer see the default store version.

ProStores includes two types of rules, built-in rules and custom rules. Built-in rules include store language/locale and default store version. The built-in rules are described in Edit Language/Locale and Edit Status, respectively.

ProStores evaluates built-in rules before evaluating custom rules.

Add Custom Rule

Click the **Add Custom Rule** button to add a rule to the store version. Click the trash icon () to remove a custom rule.

Field	Description
Evaluation Logic	<p>The AND/OR operators work as follows:</p> <ul style="list-style-type: none">◆ OR evaluates to true if any values are true.◆ AND evaluates to true if all values are true.◆ ProStores evaluates all OR rules first, followed by all AND rules. <p>For example, if you define five rules as follows:</p> <ul style="list-style-type: none">◆ Rule 1: OR◆ Rule 2: AND◆ Rule 3: OR◆ Rule 4: AND◆ Rule 5: OR <p>ProStores evaluates the rules as follows: (Rule 1 OR Rule 3 OR Rule 5) AND Rule 2 AND Rule 4. This means that ProStores will display this store if any one of Rule 1, Rule 3, or Rule 5 is true, and both Rule 2 and Rule 4 are true.</p>

Important: Working with custom rules requires in-depth knowledge of browser headers and values. For more information, see the HTTP specification at <http://www.w3.org/Protocols/>. Also, ProStores does not validate your store rules. If you define two or more store versions with the same rules, you lose control over which store your customer will see.

Custom Rule Structure

A rule definition consists of the following parts:

Field	Description
Logic Operator	Select whether the rule match is OR or AND .
HTTP Header	Enter the HTTP header to test. Examples are <code>user-agent</code> , <code>content-type</code> ,

Field	Description
	and connection.
Rule Match	The rule match list determines how ProStores will match the value.
Rule Value	Enter the value to be matched in the rule. For more information, see the HTTP specification at http://www.w3.org/Protocols/ .

The following sample rules demonstrate match options and values. The example then shows how ProStores applies each rule to a hypothetical visitor. (Note that this example uses the evaluation logic discussed in the previous section.)

Rule	Operator	HTTP Header	Option	Value
1.	OR	user-agent	Contains	MSIE 7.0
2.	AND	user-agent	Ends With	Win32)
3.	OR	user-agent	Matches Exactly	Mozilla/4.0
4.	AND	user-agent	Starts With	Mozilla
5.	OR	user-agent	Contains	MSIE

Example 1

A customer visits your store using Internet Explorer 7.0b. The customer's browser sends the following HTTP headers as part of the request:

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0b; Win32)		
Rule	Result	Reason
1.	true	The header value contains the value specified in the rule.
2.	true	The header value ends with the value specified in the rule.
3.	false	The header value does not match exactly, because the header value contains more characters than specified in the rule.
4.	true	The header value starts with the value specified in the rule.
5.	true	The header value contains the value specified in the rule.

Evaluating these rules results in **true** because the result is (Rule 1 OR Rule 3 OR Rule 5) AND Rule 2 AND Rule 4, which becomes (true OR false OR true) AND true AND true. **Only one** has to be true (one is) and all **ANDs** have to be true (they are). ProStores will display this store version to the visitor.

Example 2

A customer visits your store using Mozilla Firefox 2.0. The customer's browser sends the following HTTP headers as part of the request:

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.1) Gecko/20061204 Firefox/2.0.0.1		
Rule	Result	Reason
1.	false	The header value does not contain the value specified in the rule.
2.	false	The header value does not end with the value specified in the rule.
3.	false	The header value does not match exactly, because the header shows that the visitor is using an entirely different browser.

Rule	Result	Reason
4.	true	The header value starts with the value specified in the rule.
5.	false	The header value does not contain the value specified in the rule.

Evaluating results in `false` because the result is (Rule 1 OR Rule 3 OR Rule 5) AND Rule 2 AND Rule 4, which becomes (false OR false OR false) AND true AND false. At least one OR has to be true (none are) and all ANDs have to be true (only one is). ProStores will not display this store version to the visitor.

Edit Status

Use the Edit Status page to select the default store version that you want to display to customers. You can also use the Edit Status Page to select which stores your customers can see.

Because ProStores needs to have at least one store version available to display, you cannot turn off the default store version.

After making your selections, click the **Save** button to save your changes, or click the **Cancel** button to return to the Store Versions page.

On

This option enables you to activate or deactivate one or more store pages. Deactivated pages will not appear in your ProStores Web store.

Select the **On** check box for each store version you want available to your customers. Clearing the **On** check box makes the store version unavailable.

Default

This option lets you decide which store version is the default version. The default version is the one that ProStores will display to your customers if no other rules cause the display of another store version.

Select the **Default** option for the store version you want to use as the default store version.

Chapter 6: Advanced Design Concepts

This section covers advanced store design concepts, such as using multiple templates and working with files.

Working with Images

Images are used throughout your store to create your overall store design, as well as provide visual information regarding your products and categories.

Uploading Images

You can upload both single image files, and archived zip files. You can also upload images using the **Upload Images** wizard.

The maximum file size for uploading a single image or image archive is 10 MB. Image names may not contain spaces. In addition, archive files must be created in the standard ZIP file format.

You can upload single image files.

The maximum file size for uploading a single image is 10 MB. Image names may not contain spaces.

Category, Manufacturer and Product Images, and Favorites Icon

The Category, Manufacturer and Product images are used to describe your product catalog. The “Favorites” icon is displayed next to your url in the browser’s address bar, and in your “Favorites” menu when a page is bookmarked. To upload an image, icon or archive:

- 1 Browse to Support > Upload Images. The Upload Images page appears.
- 2 Under **Shared Images**, select the image type you are uploading from the list.
- 3 Enter the file and path name, or click **Browse** to browse to its location on your local drive.
- 4 Select the type of file you’re uploading from the list.
- 5 Click **Upload**. The image(s) will be uploaded to the appropriate directory in the Store Content Area. If you uploaded an archived zip file, the images will be expanded in the directory to which you have imported them, and the archive file will be deleted.
- 6 To automatically generate thumbnails from product images, select the **Generate thumbnail from product image** check box (only available for product images), then select the size of the thumbnails you wish to generate.

Generated thumbnail images will be in the .jpg format, which does not support transparency. If a product image is a .gif with a transparent background, the generated thumbnail .jpg will contain a solid color where the .gif was transparent.

Your “Favorites” icon must be named “favicon.ico”. In addition, its appearance in Internet Explorer is somewhat inconsistent.

Store Logo

You can specify an image that serves as your store logo. Once you have indicated that you wish to use a logo, and have uploaded it, the image will appear in the storefront.

If the image does not appear in the storefront, add the following code to the location in which you would like the logo to appear:

```
<ss:if test="$store.isFeatureEnabled('StoreLogo') ">
  <ss:image source="$storeVersion.storeLogo" border="0"/>
</ss:if>
```

To upload a store logo:

- 1 Browse to Store Design. The Store Design page appears.
- 2 Click the **Change** link next to Logo in the **Design Settings** section. The Logo page appears.
- 3 Select the **I will use a new store logo** option.
- 4 Enter the file and path name, or click the **Browse** button to browse to its location on your local drive.
- 5 Click the **Save** button. The image will appear in your store if your store theme supports a store logo image.

To remove a store logo:

- 1 Browse to Store Design. The Store Design page appears.
- 2 Click the **Change** link next to Logo in the **Design Settings** section. The Logo page appears.
- 3 Select the **I will not use a store logo** option.
- 4 Click the **Save** button. The image will no longer appear in your store.

Store Design Images

These are images that used for your general store design, such as theme images. There are several methods for uploading these images.

To upload an image from your machine using ProStores Store Design:

You must be using Internet Explorer version 5.5 or higher to use this feature.

- 1 Browse to Store Design > Page Template Manager. The Page Template Manager page appears.

- 2 Click the title link of the page template that will hold the image. The page template opens in the Page Template Editor.
- 3 Click the **Advanced Editor** tab.
- 4 Using Windows Explorer, find the image you wish to add, and then “drag” it onto the page template where the image should go. The Upload and Insert Image dialog box appears.

In newer versions of Internet Explorer, you may need to click once inside the Advanced Editor to activate the control.

- 5 Set the options for the image, such as adding alt text, and click the **Insert** button. The image appears in the template.
- 6 Click the **Save and Publish** to publish the change, or click the **Save** button to save the change without publishing it to your store.

To upload a store design image or an image archive using the Upload Images tool:


- 1 Browse to Support > Upload Images. The Upload Images page appears.
- 2 Under **Store Design Images**, enter the file and path name, or click the **Browse** button to browse to its location on your local drive.

If you are using multiple store versions, select the appropriate version from the **Store Version Name** pull-down menu.

- 3 Select the type of file to upload from the **File Type** pull-down menu.
- 4 Click the **Upload** button. If you upload an archived ZIP file, the images will be expanded in the directory to which you have imported them, and the archive file will be deleted.

You can also upload an image from another Web site using the ProStores Store Design Copy Center tool (if you are using Internet Explorer version 5.5 or higher).

To copy an image from another Web site:

- 5 Navigate to Store Design > Page Template Manager.
- 6 Open the page template that want to copy content into using one of the two methods for opening a page in the Page Template Editor.
 - ◆ Click the title of the page template.
 - ◆ Select the check box next to the page template, and then select **Edit Page Template** from the **More Actions...** pull-down menu.
- 7 Click the **Advanced Editor** tab.
- 8 Click the **Copy Center** button () on the tool bar. The Copy Center opens in a new window.
- 9 In the Copy Center window, enter the full HTTP address of the Web page from which you wish to copy content, and click the **Go** button. The Web page appears in the Copy Center window.

- 10 Use your mouse to select the content you wish to copy in the Copy Center, and then drag the content to the appropriate location in the Advanced Editor.

To make content selection easier, right-click inside the Copy Center window and clear the **Browser mode** option.

- 11 In the **Migrate Embedded Image** dialog box, select one of the two URL options:
 - ♦ **Leave the Source URL ...** inserts the image, but continues to point to the original source file (instead of uploading the image to your store).
 - ♦ **Copy the image ...** uploads the image to your store's directory on the server, and then points to that uploaded image.
- 12 (Optional) Select the check box to apply the selection to the remaining images that you copy from this Web page.
- 13 When done, close the Copy Center window.

Using and Updating Store Images

Once you've uploaded your store images, you'll want to make them available to your store.

If you have uploaded images with the same file names and pixel sizes as existing images, such as replacing banners (assuming pixel sizes are specified in the template), the image can immediately be viewed in your store (you may need to refresh the page you're currently viewing in the browser to see the change).

If you have uploaded images with the same file names but *different* pixel sizes, you'll need to edit the templates containing those images and update the width and height tags of those images.

For new images, you can add them to your templates by editing the templates in which you want to display the image. The `<ss:image source="$store.images['imagename.ext']">` tag is used to define the path to the location where your images are stored.

Using Product (Catalog) Images

To display images with your products, you'll want to enter the appropriate file name when adding or editing a product (using the Store Administration Product), and then use the appropriate ProStores tag so that image is displayed. There are two possible product images:

- 1 **Photo:** this is the larger of the two images, typically seen in the Catalog Detail template. The corresponding tag that displays this image is `<ss:image source="$product.photoImage"/>`.
- 2 **Thumbnail:** this is the smaller of the two images, and is typically seen in the Catalog List template. The corresponding tag that displays this image is `<ss:image source="$catalog.images[$product.thumbnailImageName]" border="0"/>`.

Using an Image in Place of the "Submit" Button

To have a GIF or JPEG image appear in place of the standard Submit form button, use the following HTML code instead.

```
<ss:button source="$templateSet.images['imagenamename.ext']">
```

Where "imagenamename.ext" is the name of the image. You will need to upload this image before this code will work.

Customizing Tag Images

The store makes use of several images to perform an action. These images are standard GIF files that you can customize to give your store a unique look. It is highly recommended that you make a copy of the original images before making any changes.

To make changes to these images, you can download them to your local machine using the File Management feature, and then upload them using the Upload Images feature.

Customizations to these images may be overwritten if you revert to an earlier store version. Please back up your custom images.










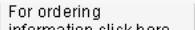


Image Name	Tag	GIF Name	Image*
Buy It	<ss:link shortcut="BuyIt"/>	buyit.gif	
Buy It Now	<ss:link shortcut="BuyItNow"/>	buyitnow.gif	
Change	<ss:button source="\$templateSet.images['change.gif']" border=0/>	change.gif	
Checkout New Customers	<ss:form shortcut="Checkout"/>	checkout.gif	
Checkout Returning Customers	<ss:form shortcut="CheckoutPro"/>	checkoutpro.gif	
Closed	<ss:image source="\$templateSet.images['closed.gif']"/>	closed.gif	
In Stock	<ss:image source="\$templateSet.images['instock.gif']"/>	instock.gif	
Join/Subscribe	<ss:image source="\$templateSet.images['join.gif']"/>	join.gif	
Lead	<ss:image source="\$templateSet.images['lead.gif']"/>	lead.gif	
Lead	<ss:form shortcut="AddToCartImage"/>	lead.jpg	

Image Name	Tag	GIF Name	Image*
Members Only	<source="\$templateSet.images['members.gif']"/>	members.gif	
Register	<ss:image source="\$templateSet.images['register.gif']"/>	register.gif	

*Some themes may come with alternatives for these images. However, the same principles for customizing those images apply.

Working with Files

In addition to the templates that ProStores provides, you can also add static pages and files to complement your store.

Uploading Files

This option is only available if it has been enabled by your Site Administrator

You can upload both single files (such as HTML pages, multimedia files, JavaScript files, style sheets, etc.), and archived zip files. Images should be uploaded using the *Uploading Images* (page 50) feature.

The maximum file size for uploading a single file or file archive is 10 MB.

To upload a file or file archive:

- 1 Browse to **Support > Upload Files**. The Upload Files page appears.
- 2 Select the file type you are uploading from the list.
- 3 Enter the file and path name, or click the **Browse** button to browse to its location on your local drive.
- 4 From the **File Type** pull-down menu, select the type of file to upload.
- 5 Click the **Upload** button. The file(s) will be uploaded to the root directory in the Store Content Area. If you uploaded an archived zip file, the files will be expanded in the root directory, and the archive file will be deleted.

Once you have added the file, you can access it using the `$store.files['FILENAME.xxx']` syntax.

Creating a Splash Page

Advanced Mode

This option allows you to create a new HTML page that search engines can find and index to better market your store. However, some search engines will *not* spider “splash” pages, so use this feature with caution.

This tool creates a very simple page that you will likely want to edit. To do so, use the Support Manager | File Management tool to download the file, make your edits, and then upload it (either using FTP, or the Upload Static Page tool) when done.

To create a static page:

- 1 Browse to **Support | Create Static Page**. The Create Static Page page appears.
- 2 Enter a file name (including the .htm or .html extension) for the new page. If you use “default.htm”, this page will overwrite the page that has links to your Store Administration area and Storefront. If you do not want to overwrite this page, use another name.
- 3 Enter a **Title** for the static page. This appears in the title bar of the browser. Some search engines use this, some don’t.
- 4 Enter a brief description of your store in the **Meta Description** field. Again, some search engines use this, while others don’t. Also, some search engines will use this text to describe your site when displaying search results. While there are no formal limits on the number of characters you can use, generally most engines will look at the first 200 characters.
- 5 Select the desired options under **Place in Meta Tags** to place the categories and/or products inside meta tags. Some search engines look in meta tags and some don’t. Including product information could result in a very large page.
- 6 Select the desired options under **Place in Page Body** to place the categories and/or products inside the main body of the page. Again, including product information could result in a very large page.
- 7 Click **Submit** to save your work and create the page.
- 8 You can download and edit the page by using the *Managing Files* (page 57) feature. The page is located in the Store Content area of the File Manager.

Once the file is created, you can browse to it by entering one of the following addresses in your browser’s address box:

- ◆ If you have a dedicated domain name (such as www.mystore.com), use “www.mystore.com/stores/[store shortname]/[filename.ext]”
- ◆ If you have a shared domain name (such as www.stores.com/mystore/), use “www.stores.com/[filename.ext]”

Cascading Style Sheets

Linked style sheets allow you to define all of your font and color information in one central location.

If you are familiar with style sheets and wish to use them to format your store:

- 1 Upload your *.CSS file to your store using the Support Manager | Upload Files feature (if it has been enabled by your Site Administrator). See *Uploading Files* (page 55) for more information.

- 2 Edit your header template(s) to contain the following code (between the `<head>` `</head>` tags:

```
<link rel="STYLESHEET" type="text/css" href="$store.files['FILENAME.css']">
```

- 3 Edit all your templates to remove the font tags and replace them with the appropriate elements and classes (such as `<p></p>`, `<h1></h1>`, etc.).

JavaScript

JavaScript allows you to add scripting functionality to your store (such as hover buttons, popup windows, etc.).

If you are familiar with JavaScript and wish to use it in your store:

- 1 Upload your *.js file to your store using the Support Manager | Upload Files feature (if it has been enabled by your Site Administrator). See *Uploading Files* (page 55) for more information.

- 2 Edit your header template(s) to contain the following code (between the `<head>` `</head>` tags:

```
<SCRIPT LANGUAGE="JavaScript" SRC="$store.files['FILENAME.js']"></SCRIPT>
```

Multimedia Files

ProStores has done some testing with including different multimedia in your pages (such as Flash files). However, we recommend that you thoroughly test including such files on different platforms and browsers.

Multimedia files should be uploaded to your store using the Support Manager | Upload Files feature (if it has been enabled by your Site Administrator). See *Uploading Files* (page 55) for more information. Use the `<ss:media/>` tag to reference your file.

Managing Files

File Management allows you to view, manage and move files located in your directory on the server.

There are three content areas:

Content Area	Description:
Store Content Area	Contains all files that can be viewed via a browser, such as static HTML pages and images.
Store Data Area	Contains all files that either cannot or should not be viewed via a browser, including configuration and log files.
FTP Area	If your Site Administrator has provided you with an FTP directory on the server that is outside of your Store Data Area, this link will provide you with access to that FTP area. See your Site Administrator about providing you with an FTP directory.

To switch from one area to another, click the link under **Switch to**.

To manage your server files:

- 1 Browse to **Support > File Management**. The File Management page appears.
- 2 To view a file or image, click its link. You can also “right-click” the link to download the file.
- 3 To open a folder, click its link.
- 4 To go back to the folder containing the current folder, click the **Parent Directory** link.
- 5 To delete a folder or file, select the folder or file and click **Delete**. Deleting a folder will also delete any folders and files contained within it.
- 6 To copy a folder or file:
 - ◆ Select the check box of the folder or file and click **Copy**. The folder(s) and file(s) are placed on the clipboard.
 - ◆ Go to the folder into which you want to copy the folder or file and click **Paste**.
- 7 To move a folder or file:
 - ◆ Select the check box of the folder or file and click **Cut**. The folder(s) and file(s) are removed from their current location and are placed on the clipboard.
 - ◆ Go to the folder into which you want to move the folder or file and click **Paste**.
- 8 To download a file:
 - ◆ Right-click on the file you want to download. A context menu appears.
 - ◆ Select “Save Target As...” or “Save Link Target As”. A Save As window appears.
 - ◆ Browse to the location to which you wish to save the file and click **Save**. The file is saved to the specified location.

Omitting the Header and Footer

ProStores allows you to prevent the Header and Footer templates from displaying on a page by page basis. This allows you to achieve custom designs for individual templates. For example, if you want your storefront home page to have a different look than the rest of your templates, you can omit the Header and Footer from that template.

To do so, simply add a `<HTML>` tag as the first item on the appropriate template, and a `</HTML>` template as the last item. Then add your custom design to the template. In the example of creating a custom design for your home page, edit the Storefront template to include an `<HTML>` tag at the beginning, and an `</HTML>` tag at the end. Then add your custom design, and publish the template. Your home page now appears without the Header and Footer template.

Customizing the Display of the Category List

By default, the themes that are shipped with ProStores will display all active categories, and all subcategories, in the store, with links to the search results for that category. This list is displayed on either the header or the footer of the store, and on some home page layouts.

You can control the “depth” of this list from within Store Administration, on the **Product | Categories** landing page. You can display:

- ◆ Top-level categories only.
- ◆ All categories and subcategories (default).
- ◆ Subcategories up to a specified level.

These settings only affect the use of the \$catalog.categoryList member. Any other members that display categories are not affected.

Using Multiple Templates

ProStores gives you the option of using different versions of the same template, depending on the information you are trying to display. Using different templates based on a category, product or manufacturer allows you to customize the look and feel of your store right down to the product detail level.

For example, if you have a pet store, you may decide to use a different background pattern and banner when displaying catalog lists for cat related products than for dog related products.

To do so:

- 1 Create unique header templates for each category (header templates are usually where banners and background patterns are defined).
- 2 Define the categories “Cats” and “Dogs”, assigning the appropriate header templates to each category.
- 3 Make sure you assign a category to each Cat and Dog related product.

Where You Can Use Multiple Templates

There are several different criteria for which you can define the use of different templates: Category, Product and Manufacturer.

Using multiple templates based on Category allows you to completely change the look of your catalog listing based upon the category that the customer chose (either by clicking a link you provided, or by selecting a category on your search page).

- ◆ The Header template allows you to change the body color and background, header banner, and navigation.

- ◆ The Catalog List template allows you to change how the listing of products will appear.
- ◆ The Footer template allows you to change the appearance of any navigation you have along the bottom of your pages, and completes the code that was started in the Header template.

Using multiple templates based on product allows you to change the layout for each product detail page, as well as define a different order extension template for each product.

- ◆ The Catalog Detail template affects the look and feel of the page that contains detailed information regarding this product.
- ◆ The Order Extensions template allows you to customize different order extension layouts for each product.
- ◆ If some of your products use different attribute information than Color and Size, the Cart Add template can be customized to display the appropriate attribute labels. If you create a custom Cart Add template, you'll also need to create a custom Catalog Detail template that refers to your new Cart Add template.

Using multiple Catalog List templates based on manufacturer allows you to display unique catalog lists when displaying search results by manufacturer.

Using multiple Supplier templates based on supplier allows you to customize the notifications (fax, EDI) you send for each supplier.

How to Create and Use Multiple Templates

To create and use multiple templates:

- 1 Add a new template by copying an existing template. See *Import and Replace Page Template* (page 33) for more information.
- 2 Open the page in your favorite HTML editor and customize the template.
- 3 Import the template. See *Export Page Template* (page 33) for more information. Since this is a new template, be sure to add a **Name** and **Description** for the template, and select a template **Type**.
- 4 After you import the template, it will appear in the appropriate lists.
 - ◆ Header and Footer templates will appear on the Category page.
 - ◆ Catalog List templates will appear on the Category and Manufacturers pages.
 - ◆ Order Extensions templates will appear on The Product Manager page.
 - ◆ Supplier templates will appear on the Supplier Profile page.
 - ◆ Catalog Detail templates will appear on the Product Profile.
- 5 If you created a custom Cart Add template, you'll also need to create a custom Catalog Detail template that refers to your new Cart Add template.

User Defined Templates

User defined templates allow you to quickly add pages to your store without using the Static Page feature. User defined templates are displayed with the store's header and footer, and can be used for content such as adding pages to your "About Us" area, or additional Customer Service information.

To add a user defined template:

- 1 Create your new template.
- 2 Import the template. See *Import and Replace Page Template* (page 33) for more information. Since this is a new template, be sure to add a name and description for the template, and select *User Defined Template* from the **Type** list.
- 3 Once the template has been imported, you can create a link to it from any page in your store using the `<ss:link source="$store.files['TemplateName']"/>` tag.

To delete a user defined template, click the delete icon next to the template name in Page Template Manager.

Only user defined templates may be deleted.


Frequently Asked Questions: Store Design

The following are common design questions and answers.

How do I change the background colors used in a theme?


While ProStores offers a variety of color combinations in its themes, you may want to customize the theme you've chosen by changing one or more of its background colors. There are three methods for changing the background color - the method you will use depends upon the theme you have selected, and the design of your store.

If you want to change the background color of the body of the pages in your store:

- 1 Browse to Store Design > Design Settings.
- 2 Select the appropriate store version from the **Store Version** pull-down menu.
- 3 Click the Fonts and Colors **Change** link. The Fonts and Colors page opens.
- 4 The **Background** field in the **Colors** section displays the current color value and a preview of the color. You can change the color in any one of three ways:
 - ◆ Enter the name of the color in the **Background** field. For example, "blue".
 - ◆ Enter the hexadecimal value of the color in the **Background** field in the format **#RRGGBB**, where RR is the value for red, GG for green, and BB for blue.
 - ◆ Click the color selector button () to select from a color palette.

- 5 (Optional) Click the **Preview** button to see how the new color will look in your store.
- 6 Save your work.
 - ◆ Click the **Save** button to save your changes but not publish them.
 - ◆ Click the **Save and Publish** button to save your changes and publish them to your store.
 - ◆ Click the **Cancel** button to discard your changes.

If you want to change the background color in a specific table cell for a specific page template, and you are using Internet Explorer version 5.5 or higher:

- 1 Browse to Store Design > Page Template Manager.
- 2 Select the appropriate store version from the **Store Version** pull-down menu.
- 3 Click the title link for the page template you want to change. The Page Template Editor appears.
- 4 Click the **Advanced Editor** tab.
- 5 Place your cursor in the cell whose background color you wish to change.
- 6 Click the right mouse button. The context menu appears.
- 7 Select **Table Properties...** option. The Table Properties dialog box appears.
- 8 Enter the new color in the **Background Color** field, or click  to select a color.
- 9 Click the **OK** button. The background color of the table cell changes to match your selection.
- 10 (Optional) Click the **Preview** button to see how the new color will look in your store.
- 11 Save your work.
 - ◆ Click the **Save** button to save your changes but not publish them.
 - ◆ Click the **Save and Publish** button to save your changes and publish them to your store.
 - ◆ Click the **Cancel** button to discard your changes.

If neither of these methods work as you expected, or don't fit your situation, you can try editing the Header page template source code as follows.

These steps detail editing the Header page template source code. Because the Header page template stores the background color for your entire store, changing the background color here changes it for all pages. You can also use this procedure to change the background color of individual page templates.

CAUTION: Edit the source code with care, as errors could render your store unusable. Source code editing is only recommended for those who are comfortable working directly with HTML and SSML.

- 1 Browse to Store Design > Page Template Manager.
- 2 Select the appropriate store version from the **Store Version** pull-down menu.

- 3 Click the title link for the Header page template. The Page Template Editor appears.
- 4 Click the **Advanced Editor** tab.
- 5 (Optional) Select the **Wrap Lines** check box for easier reading of the code.
- 6 Look for either the `background-color` or `bgcolor` attribute, and replace the value (the part after the equals sign) with your new color values. You can use your browser's **Find** or **Search** command to quickly find these tags.
- 7 (Optional) Click the **Preview** button to see how the new color will look in your store.
- 8 Save your work.
 - ◆ Click the **Save** button to save your changes but not publish them.
 - ◆ Click the **Save and Publish** button to save your changes and publish them to your store.
 - ◆ Click the **Cancel** button to discard your changes.


How do I add my company logo to my store?

The Store Design Logo page makes adding a company logo quick and easy.

- 9 Browse to Store Design.
- 10 Select the appropriate store version from the **Store Version** pull-down menu.
- 11 Click the **Change** link next to Logo in the **Store Design** section. The Logo page appears.
- 12 Select the **I will use a new store logo** option.
- 13 Click the **Browse...** button and navigate to your store logo image.
- 14 Click the **Save** button.

How do I undo a mistake I made when working on my templates?

ProStores Store Design offers several opportunities for you to undo your work.

If you...	...then do this
Have <i>not</i> saved or published your changes:	Use your operating system's or browser's Undo command. In Windows, this is usually CTRL + Z . In certain browsers, you can find the Undo command on the Edit menu.
Have <i>saved</i> your changes, but haven't published them:	Click the Undo button on the toolbar () or at the bottom of the Page Template Editor. This discards all changes since the last time you <i>saved</i> the template.
Have <i>published</i> your changes:	If you've already published your changes, you'll need to use the Revert to Baseline feature. However, this will undo all changes you've made to <i>all</i> templates since the last time you set the baseline.

See Also

- ◆ *Revert to Baseline* (page 21)
- ◆ *Revert to Original Version* (page 21)

I'm using multiple store versions. How do I redesign a specific store version for my store?

Store Design provides all the tools you need to change the design for any store version.

If you need to modify the theme for a store version, you may want to copy the store version to a new version, then work in the copy until you are satisfied with the design. You may then either keep or delete the original version.

15 (Optional) Create a copy of the original store version:

- ◆ Browse to Store Design > Store Versions.
- ◆ Select the check box next to the store version that you want to copy.
- ◆ Select the **Copy Store Version** action from the **More Actions...** pull-down menu. The Add Store Version page appears.
- ◆ Enter a name for the store version in the **Store Version Name** field.
- ◆ (Optional) Select a language and location from the **Language/Locale** pull-down menu.
- ◆ Click the **Copy** button.

16 Browse to Store Design > Page Template Manager.

17 Select the appropriate store version from the **Store Version** pull-down menu.

You are now ready to modify any of the store design settings and options.

How do I change the fonts used in my store?

To make changes to your fonts:

18 Browse to Store Design > Design Settings.

19 Select the appropriate store version from the **Store Version** pull-down menu.

20 Click the Fonts and Colors **Change** link. The Fonts and Colors page opens.

21 Select appropriate fonts and colors in the **Fonts** section.

22 (Optional) Click the **Preview** button to see how the new fonts and colors will look in your store.

23 Save your work.

- ◆ Click the **Save** button to save your changes but not publish them.
- ◆ Click the **Save and Publish** button to save your changes and publish them to your store.
- ◆ Click the **Cancel** button to discard your changes.

See Also

- ◆ Fonts and Colors

How can I create a new design for my store while leaving the existing design as it is?

Sometimes you may need to redesign your store, but you don't want affect your existing design while doing so. Store versions offer one solution. By creating a new store version for your redesigned store, you can change the design while leaving the existing design untouched.

To create a copy of a store version:

- 1 Browse to Store Design > Store Versions.
- 2 Select the check box next to the store version that you want to copy.
- 3 Select the **Copy Store Version** action from the **More Actions...** pull-down menu. The Add Store Version page appears.
- 4 Enter a name for the store version in the **Store Version Name** field.
- 5 (Optional) Select a language and location from the **Language/Locale** pull-down menu.
- 6 Click the **Copy** button. The new version appears in the **Store Versions** list in the Store Versions page.

You now have a copy of your store design that you can work with while leaving your original design untouched. After you make your design changes, you may keep the original store or delete it.

Chapter 7: Code Snippets

This chapter contains snippets of SSML and HTML code that show the power and flexibility of ProStores.

Support Multiple Featured Products and Categories

This topic presents sample code for supporting multiple Products and multiple featured categories on your storefront.

Versions of ProStores later than 7.0 support multiple featured products and multiple featured categories. Each store can have up to five featured products and up to five featured categories.

See *Using the Template Source Editor* (page 42) for more information.

Example: Multiple Rotating Featured products

To display multiple rotating featured products, your Storefront template must contain the following code:

```
<ss:set name="featuredProducts" value="$store.products.getRandomFeaturedProduct()" />
<ss:if test="$featuredProducts">
  <ss:link source="$featuredProducts"><ss:value
source="$featuredProducts.name" /></ss:link>
</ss:if>
```

Example: Featured product Thumbnail Image

The following example shows how to display the multiple rotating featured product thumbnail image, name, and description:

```
<ss:set name="featuredProducts" value="$store.products.getRandomFeaturedProduct()" />
<ss:if test="$featuredProducts">

  <ss:link source="$featuredProducts"><ss:image
source="$featuredProducts.thumbnailImage" alt="$featuredProducts.name"
border="0" /></ss:link><br>
  <ss:link source="$featuredProducts"><ss:value
source="$featuredProducts.name" /></ss:link><br>
  <ss:value source="$featuredProducts.description" />

</ss:if>
```

Example: Display All Featured products

The following example shows how to display all of the featured products at the same time:

```
<ss:set name="featuredProducts" value="$store.products.getFeaturedProducts()" />
<ss:if test="$featuredProducts">
  <ss:foreach item="product" within="$featuredProducts">
    <ss:if test="$product.thumbnailImage">
      <ss:link source="$product">
```

```

        <ss:image source="$product.thumbnailImage" alt="$product.name"
border="0"/></ss:link>
        <br>
    </ss:if>
    <ss:link source="$product"><ss:value source="$product.name"/></ss:link>
    <br>
    <ss:value source="$product.brief"/>
    <br><br>
</ss:foreach>
</ss:if>

```

Example: Display Featured Categories

The following example shows how to display the featured categories on your storefront:

```

<ss:set name="featuredCats" value="$store.categories.getRandomFeaturedCategory()"/>
<ss:if test="$featuredCats">
    <ss:link source="$featuredCats"><ss:value source="$featuredCats.name"/></ss:link>
</ss:if>

```

Support Advanced Fonts and Colors

ProStores provides support for copying your eBay Store font styles and color theme settings to your ProStores Web store theme if you have granted permission to ProStores to access your eBay Store in the eBay Setup page.

If your store template does not support eBay font styles and color scheme settings, you must add some code to your store version in order for your eBay theme settings to appear in your ProStores Web store. Each tag listed below corresponds to a setting in the Fonts and Colors page, which you can access from the Design Settings page.

This topic presents tags and examples in three sections:

- ◆ *Advanced Fonts and Colors Tags* (page 67) presents a table of tags that add support for advanced fonts and colors, together with a description of each tag.
- ◆ *Example: Modify Styles* (page 68) shows how to add advanced fonts and colors support to the Header stylesheet.
- ◆ *Example: Modify Template* (page 68) shows how to add advanced fonts and colors support to the page template HTML code.

Advanced Fonts and Colors Tags

Store Name tags add support for applying advanced fonts and colors to your store name.

Tag	Description
\$font.storeName.face	Font family used to display the store name.
\$font.storeName.size	Relative font size used for HTML tags.
\$font.storeName.color	Text color.
\$font.storeName.pixelSize	Font size in pixels, for use in style sheets.

Section Title tags add support for applying advanced fonts and colors to the section headings on your store pages.

Tag	Description
\$font.sectionTitle.face	Font family used to display the store name.
\$font.sectionTitle.size	Relative font size used for HTML tags.
\$font.sectionTitle.color	Text color.
\$font.sectionTitle.pixelSize	Font size in pixels, for use in style sheets.

Background Color tags add support for applying advanced color schemes to your store.

Tag	Description
\$templateSet.colorScheme.background	Background color.
\$templateSet.colorScheme.accent	Accent color.
\$templateSet.colorScheme.primary	Primary color.
\$templateSet.colorScheme.secondary	Secondary color.

Example: Modify Styles

The example code in this section goes in the Header Source between the stylesheet opening and closing tags.

The following example shows how to apply the primary color as the background color for a style class:

```
.primary {
  background-color:<ss:value source=$templateSet.colorScheme.primary/>;
}
```

The following example shows how to apply the store name font and color to a style class:

```
.storename {
  font-family:<ss:value source=$font.storeName.face/>;
  color:<ss:value source=$font.storeName.color/>;
  font-size:<ss:value source=$font.storeName.pixelSize/>px;
}
```

Example: Modify Template

The example code in this section goes in the HTML source for a store page template.

The following example shows how to apply the primary color as the background color for a table:

```
<table border="0" bgcolor="$templateSet.colorScheme.primary">
```

The following example shows how to apply the section title font settings to a section title.

```
<ss:font source="$font.sectionTitle">Section Title</ss:font>
```

After you add these tags, you can change their values in the Fonts and Colors page.

Adding Multiple Products to a Cart

You can create a Catalog List template that will allow your customers to select multiple products and add them to their cart with one click. To do so, insert the following code into the Catalog List template:

```
<table border="0" cellpadding="2" cellspacing="4">
<tr valign="bottom" bgcolor="indigo">
  <td align="center">Check To<br>Order</td>
  <td>Product</td>
  <td align="right">Our Price</td>
</tr>
<ss:form shortcut="CartStart">
  <ss:foreach item="listing" within="$catalog.results">
    <ss:foreach item="product" within="$listing.products">
      <tr valign="top">
        <td align="center">
          <ss:if test="$product.hasAttributes != true">
            <ss:checkbox name="cartchk" value="$product.productNumber"/>
          </ss:if>
        </td>
        <td><ss:link source="$product"/></td>
        <td align="right"><ss:value source="$product.displayPrice"
format="money"/></td>
      </tr>
    </ss:foreach>
  </ss:foreach>
</table>
<p><ss:button type="submit" value="Add To Shopping Cart"/></p>
</ss:form>
```

Quick Search Form

You can create a mini keyword search form that can appear on any template (such as on the header template), so that your registered customers don't have to go to the Member Login template. To do so, insert the following code on the desired template:

```
<form method=post action="$templateSet.searchResults">
<input type="text" name="keyword" size=8>
<ss:button source="$templateSet.images['imagename.gif']" alt="Search" border="0"/>
</form>
```

Where “imagename” refers to the name of a “go” image you have uploaded to your store. If you wish to use a standard form button instead of an image, use `<ss:button type="submit" value="Submit"/>`

You may also add font and table formatting as needed.

UPS Tracking Link

You can add a link to the Invoice template that will take your customers directly to the UPS Package Tracking site, and display tracking information for a specific order. To do so, edit the appropriate template (such as Invoice or Customer Service) and add the following code:

The values in this form may change if UPS makes changes to its tracking application.

```
<form method="post" target="_blank"
action="http://wwwapps.ups.com/etracking/tracking.cgi">
  <input type="hidden" name="InquiryNumber1" value="$invoice.shipTrackNumber">
  <input type="hidden" name="TypeOfInquiryNumber" value="T">
  <input type="hidden" name="AcceptUPSLicenseAgreement" value="yes">
  <input type="submit" name="submit" value="Track Package">
</form>
```

Collapsible Category Lists

The following code, placed on the Header or Footer template, will create a list of categories that will “expand” to include a list of subcategories when a link is clicked.

```
<ss:foreach item="category" within="$catalog.categoryList()">
  <ss:if test="$category.treeLevel == '1'">
    <br><ss:link source="$category"/><br>
  <ss:else/>
    <ss:if test="$request.getParameter(category) == null">
      <ss:foreach item="i" within="$system.iterator($category.treeLevel)">
        &nbsp;
      </ss:foreach>
      <ss:link source="$category"/><br>
    <ss:else/>
      <ss:set name="clickedcat"
        value="$catalog.categories.getItem($request.getParameter(category))"/>
      <ss:if test="clickedcat">
        <ss:if test="$clickedcat.parent.name == $category.parent.name OR
$clickedcat.name == $category.parent.name">
          <ss:foreach item="i" within="$system.iterator($clickedcat.treeLevel)">
            &nbsp;
          </ss:foreach>
          <ss:link source="$category"/><br>
        </ss:if>
      </ss:if>
    </ss:if>
  </ss:if>
</ss:foreach>
```

Product Price Formatting

(Advanced Edition only)

By default, ProStores uses the `<ss:value macro="attributedetailfull"/>` to display all product and attribute pricing information. If you want to change the way in which that information is displayed, you can replace that macro with the following code, and then make the desired modifications.

```
<ss:foreach item="productprice" within="$product.finalPriceList">
  <ss:choose>
    <ss:when test="$counter.totalRows == 1">
      <ss:value source="$productprice"/>
    </ss:when>
    <ss:otherwise>
      <ss:choose>
        <ss:when test="$counter.isLast">
          <font color="#AA0000"><ss:value source="$productprice"/></font>
        </ss:when>
        <ss:when test="$counter.currentRow == 1">
          <s><ss:value source="$productprice"/></s>
        </ss:when>
      <ss:otherwise>
```

```

        <s><font color="#AA0000"><ss:value source="$productprice"/></font></s>
      </ss:otherwise>
    </ss:choose>
  </ss:otherwise>
</ss:choose>
</ss:foreach>
<ss:foreach item="promotion" within="$product.activePromotions">
  <ss:set name="detailText" value="$product.promotionDetailText($promotion)"/>
  <ss:if test="$detailText != null">
    &nbsp;<ss:value source="$detailText"/>
  </ss:if>
</ss:foreach>
<br>

<ss:if test="$product.hasAttributes">
  <ss:foreach item="attribute" within="$product.attributes">
    <ss:foreach item="price" within="$attribute.finalPriceList">
      <ss:choose>
        <ss:when test="$counter.totalRows == 1">
          <ss:value source="$price"/>
        </ss:when>
        <ss:when test="$counter.isLast">
          <font color="#AA0000"><ss:value source="$price"/></font>
        </ss:when>
        <ss:when test="$counter.currentRow == 1">
          <s><ss:value source="$price"/></s>
        </ss:when>
        <ss:otherwise>
          <s><font color="#AA0000"><ss:value source="$price"/></font></s>
        </ss:otherwise>
      </ss:choose>
    </ss:foreach>
    <ss:foreach item="promotion" within="$product.activePromotions">
      <ss:set name="detailText" value="$product.promotionDetailText($promotion)"/>
      <ss:if test="$detailText != null">
        &nbsp;<ss:value source="$detailText"/>
      </ss:if>
    </ss:foreach>
    <ss:value source="$attribute.attributeText"/><br>
  </ss:foreach>
</ss:if>

```

Hide Attribute Information When Not In Stock

(Advanced Edition only)

This code can be used on the Catalog Detail Template to prevent the product attribute size information from displaying when the attribute quantity is 0. It may be modified to do the same for the color attribute.

```

<td>
  <font face="arial,helv" size="2"><b>Size</b>&nbsp;<ss:foreach item="attr" within="$product.attributes">
    <ss:if test="$attr.quantity < 0">
      <option value="$attr.size"><ss:value source="$attr.size"/></option>
    </ss:if>
  </ss:foreach>
</select>
</font>
</td>

```

Order Extension Information on the Invoice Email

To display Order Extension information on the Invoice - Email template, add the following code somewhere between the `<ss:foreach item="detail" within="$invoice.details">...</ss:foreach>` tag.

```
<ss:foreach item="extensions" within="$detail.extensions">
  <ss:value source=$extensions.extensionText1/>
  ...
</ss:foreach>
```

Suppressing A Shipping Method

By default, the Merchant-Defined Shipper expects the merchant to use the three available shipping methods provided by the Merchant-Defined Shipper. If you wish to use only one or two, you can modify the code as described below.

Checkout and Checkout New Customer Templates

Delete the following code...

```
<p>
  <b>Shipping Method</b><br>
  <ss:foreach item="shipQuote" within="$cart.shippingQuotes">
    <ss:radio name="shipme" source="$shipQuote.shipMethod"
checked="$counter.currentRow==1"/>&nbsp;&nbsp;&nbsp;
    <ss:value source="$shipQuote.name"/> (<ss:value source="$shipQuote.amount"
format="money"/>)
    <br/>
  </ss:foreach>
</p>
```

...and replace it with this code (removing the option you do *not* wish to offer):

```
<p>
  <b>Shipping Method</b><br>
  <input type="radio" name="shipme" value="S" checked>&nbsp;&nbsp;&nbsp;US Postal Service Ground
($5.105)<br/>
  <input type="radio" name="shipme" value="2">&nbsp;&nbsp;&nbsp;US Postal Service Priorit
($7.95)<br/>
  <input type="radio" name="shipme" value="N">&nbsp;&nbsp;&nbsp;Next Day ($14.95)<br/>
</p>
```

Invoice Template

Replace...

```
<ss:include template="selectshipping"/>
```

...with

```
<ss:value source="$cart.shipping"/>
```

Subcategory Drill-Down on Catalog List

The following code modifies the Catalog List template to do one of the following:

- ◆ If the user clicks a category link and that category has subcategories, those subcategories will be displayed on the Catalog List template as links.
- ◆ If the user clicks a category or subcategory link that does *not* have any categories, the Catalog List will display a list of product search results.

```

<ss:set name="viewedcategory" value="$request.getParameter('category')"/>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td align="center" valign="bottom" class="mainlinks"></td>
</tr>
<tr>
<td align="center" valign="bottom" class="mainlinks">
<ss:set name="NOsubcat" value="1"/>
<ss:foreach item="category" within="$catalog.categoryList()">
<ss:if test="$viewedcategory">
<ss:if test="$category.parentchaindisplay == $viewedcategory">
<ss:foreach item="subcat" within="$category.childrenToDepth(1)">
&nbsp;&nbsp;&nbsp;<ss:link source="$subcat"/><br>
<ss:set name="NOsubcat" value="$subcat"/>
</ss:foreach>
</ss:if>
</ss:if>
<ss:else/>
<ss:if test="$category.treeLevel == '1'">
<ss:else/>
<font size="-2">&nbsp;&nbsp;&nbsp;<ss:link source="$category"/></font><br>
</ss:if>
</ss:if>
</ss:foreach>
<ss:if test="$NOsubcat == '1'">
<ss:foreach item="listing" within="$catalog.results">
<ss:sortBy item="product"/>
<ss:foreach item="product" within="$listing.products">
<tr valign="top">
<td nowrap>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>
<ss:font source="$font.paragraph">
<ss:link source="$product">
<ss:image source="$catalog.images[$product.thumbnailImageName]"
border="0"/>
</ss:link>
</ss:font>
</td>
<td width="100%">
<ss:font source="$font.paragraph">
<ss:link source="$product">
<b><ss:value source="$product.name"/></b>
</ss:link>
<br>
<ss:value source="$product.brief"/><br>
</ss:font>
</td>
<td nowrap align="left">
<ss:font source="$font.paragraph">
<b>Price</b>
<br>
<ss:include macro="productprice"/>
</ss:font>
</td>
<td nowrap>
<ss:font source="$font.paragraph">
<ss:link shortcut="BuyIt"/>
<br><br>
<ss:link shortcut="BuyItNow"/>
</ss:font>
</td>
</tr>
</tr>
<td>&nbsp;&nbsp;&nbsp;</td>

```

```

        <td colspan="4"><hr noshade></td>
      </tr>
    </ss:foreach>
  </ss:foreach>
</ss:if>
</td>
</tr>
</table>

```

Comparison Category List

The following code, placed on a custom Catalog List template, will allow your visitors to compare products within a category.

Main Catalog List Template

```

<form method="GET" action="$templateSet.templates['prodcompare']">
  <input type="hidden" name="template" value="prodcompare">
  <input type="submit" name="action" value="Compare"/>
  <ss:set name="productCount" value="$system.getMath(1)"/>

  <ss:foreach item="listing" within="$catalog.results">
    ....
    <ss:foreach item="product" within="$listing.products">
      <input type="checkbox" name="$system.concat('p',$productCount.get())"
value="$product.productNumber"/>
      <ss:directive source="$productCount.set($productCount.add(1))"/>
      [Display usual product info here]
    </ss:foreach>
  </ss:foreach>
</form>

```

Custom Catalog List Template [short name: prodcompare]

This template should have the short name specified in the <form> tag above.

```

<p><ss:font source="$font.heading">Product Comparison</ss:font></p>
<ss:set name="selCount" value="$system.getMath(0)"/>

<ss:if test="$request.getParameter('p1')">
  <ss:directive source="$selCount.set($selCount.add(1))"/>
  <ss:set name="p1"
value="$store.products.getProductWithNumber($request.getParameter('p1'))"/>
</ss:if>

<ss:if test="$request.getParameter('p2')">
  <ss:directive source="$selCount.set($selCount.add(1))"/>
  <ss:set name="p2"
value="$store.products.getProductWithNumber($request.getParameter('p2'))"/>
</ss:if>

<ss:if test="$request.getParameter('p3')">
  <ss:directive source="$selCount.set($selCount.add(1))"/>
  <ss:set name="p3"
value="$store.products.getProductWithNumber($request.getParameter('p3'))"/>
</ss:if>

[Repeat for the number of products that you support on a single page
of search results.]

<ss:if test="$selCount.get() > 0">
  <p>You selected <ss:value source="$selCount.get()"/> products to compare.</p>
  <table>
    <tr>
      <td>Product name:</td>
      <ss:if test="$p1"><td><ss:value source="$p1"/></td></ss:if>

```

```

        <ss:if test="$p2"><td><ss:value source="$p2"/></td></ss:if>
        <ss:if test="$p3"><td><ss:value source="$p3"/></td></ss:if>
        <!-- And so on for max possible products that can be selected per page    </tr>
    <tr>
        <td>Description:</td>
        <ss:if test="$p1"><td><ss:value source="$p1.description"/></td></ss:if>
        <ss:if test="$p2"><td><ss:value source="$p2.description"/></td></ss:if>
        <ss:if test="$p3"><td><ss:value source="$p3.description"/></td></ss:if>
    </tr>
    <tr>
        <td>Price:</td>
        <ss:if test="$p1"><td><ss:value source="$p1.finalPrice"/></td></ss:if>
        <ss:if test="$p2"><td><ss:value source="$p2.finalPrice"/></td></ss:if>
        <ss:if test="$p3"><td><ss:value source="$p3.finalPrice"/></td></ss:if>
    </tr>

    [And so on... for each data element that's relevant to compare.
    You can also use custom fields and attributes to compare.]

</table>
<ss:else/>
    <p>Please select 2 or more products to compare and try again.</p>
</ss:if>

```

Display Related Products on Cart Template

(Advanced Edition only)

The following code, added to the Cart template, will display a list of products related to those in a customer's cart (Advanced Edition only)

```

<ss:if test="$cart.Details">
    <p>You may also be interested in the following products:</p>

    <ul>
    <ss:foreach item="detail" within="$cart.Details">
        <ss:foreach item="relatedproduct" within="$detail.relatedProducts">
            <li><ss:link source="$relatedproduct"><ss:value
source="$relatedproduct.name"/></ss:link></li>
        </ss:foreach>
    </ss:foreach>
    </ul>
</ss:if>

```

Show Only Attributes That Are On Sale

(Advanced Edition only)

This code can be used on the Catalog Detail Template to only display those attributes that are on sale.

```

<ss:set name="previousColor" value="null"/>
<ss:foreach item="productattr" within="$product.attributes">
    <ss:if test="$previousColor != $productattr.color">
        <ss:if test="$productattr.basePrice != $productattr.finalPrice">
            <ss:foreach item="attrPrice" within="$productattr.finalPriceList">
                <ss:if test="$attrPrice != $productattr.finalPrice">
                    <ss:set name="previousColor" value="$productattr.color"/>
                    <ss:if test="$attrPrice != $productattr.basePrice">
                        <font color="#AA0000"><s><ss:value source="$attrPrice"/></s></font>
                    <ss:else/>

```

```

        <s><ss:value source="$attrPrice"/></s>
      </ss:if>
    <ss:else/>
    <font color="#AA0000"><ss:value source="$attrPrice"/></font>
  </ss:if>
</ss:foreach>
<ss:value source="$productattr.size"/><ss:value source="$productattr.color"/>
<br>
</ss:if>
</ss:if>
</ss:foreach>

```

Total Number of Items in a Cart

This code can be used on the display the total number of items that are in a cart. For example, if the cart contains five of one line item, and three of another line item, this code would display “8”.

```

<ss:set name="math" value="$system.getMath('0')"/>
<ss:foreach item="detail" within="$cart.details">
  <ss:directive source="$math.set($math.add($detail.quantity))"/>
</ss:foreach>
Total quantity: <ss:value source="$math.get()"/>

```

If you use this code on the Cart template, you can just add the `<ss:set>` tag before the existing `<ss:foreach>`, add the `<ss:directive>` tag anywhere within the `<ss:foreach>`, and then add the Total Quantity line to the end (after the `</ss:foreach>`).

Chapter 8: Testing Your Design

Once you've edited your templates, it's time to test your design.

This chapter makes testing recommendations, and provides instructions to help you test your store.

Equipment and Software to Test

To ensure that your store can be viewed and easily used by the maximum number of customers, we recommend that you test your system using a variety of software and platforms, including (but not limited to), the following:

- ◆ Operating Systems: such as Windows XP, Windows ME, and Macintosh. Don't assume that if your design works on one operating system, it will work in all versions of that system.
- ◆ Browsers: at a minimum, you should test using Internet Explorer versions 5 and 6, and Netscape versions 6 and 7. However, there are still customers with older browsers that you may want to consider testing.
- ◆ Monitor Resolution: the most common monitor resolution is 800x600. We recommend setting your monitor resolution to the minimum you want to require of your customers when testing - refer to the software that comes with your operating system for more information on setting monitor resolution.
- ◆ JavaScript: some customers disable JavaScript in their browsers for security reasons. If you have added JavaScript to your pages, or if you are using the columnar Catalog List layouts provided by ProStores, we recommended disabling JavaScript in your test browser and testing your store to ensure that it is still usable.

How to Test

Following are recommendations on how to test your store.

Testing Store Information Templates

Store Information is typically contained on the following templates. To test the design of these templates, you can either navigate to them from the storefront, or use the following URL structure:

`http://[storeurl]/Page.bok?template=[templatename]`

...where [templatename] is the template short name.

- ◆ About Us [about]

- ◆ Policies [policy]
- ◆ Store Locator [locator] (optional template)
- ◆ FAQ [faq] (optional template)
- ◆ Privacy Statement [privacy] (optional template)

Testing the Product Catalog Templates

Following are some scenarios to test your product catalog design:

- ◆ Do a product search. Does the Catalog List appear as expected?
- ◆ View the detail for any product. Does the page appear as expected?
- ◆ View the detail for any product that has color or size attributes. Does the “add to cart” portion of the page appear as expected?
- ◆ View the detail for any product that has related products. Does the “Related Products” portion of the page appear as expected? (*Advanced Edition only*)
- ◆ If applicable, search for a product that uses a custom Catalog Detail template. Does the page appear as expected?

Testing the Checkout Templates

Following are some scenarios to test your checkout-related pages.

- ◆ Purchase a product and check out as a new customer. Do the templates appear as expected?
- ◆ If applicable, purchase a product that uses an Order Extensions template, and check out. Does the Order Extensions template appear as expected? (*Advanced Edition only*)
- ◆ Purchase a product and check out, this time opting to ship this product as a gift. Does the Ship To - Gift template appear as expected? (*Advanced Edition only*)

Testing Membership and Subscription Templates

(*Advanced Edition only*)

Following are some scenarios to test your membership and subscription templates.

If you are setting up a store subscription program:

- ◆ In the storefront, is the link to “Join Now” visible or easily accessible?
- ◆ Click the “Join Now” link, and purchase the store subscription product. Do the templates appear as expected?

If you are setting up a membership area for your store:

- ◆ In the storefront, are the “Logon” and “Logoff” links visible or easily accessible?

- ◆ Register as a new customer. Do all the templates appear as expected?
- ◆ Log in as the customer you just registered as. Do any “members-only” customizations you have made appear?
- ◆ If you have customized your members-only templates, search for a product and view the Catalog List. Then, view the product detail. Do the templates appear as expected?

Testing Customer Service Templates

Your customer service templates are key to caring for your customers after they have purchased products. The following scenarios will help test the design of your customer service templates.

- ◆ Go to the Service page, and click the **Account Information** link. Does the Customer History template appear as expected?
- ◆ Go to the Service page, click the **Lost Passwords** link, and complete the form. You should receive your password via e-mail. Is the e-mail formatted as expected?

If you have selected any of the notification options in **Mail Preferences**:

- ◆ Check out as a new customer, using your own e-mail address. You should receive an order confirmation e-mail. Is the e-mail formatted as expected?
- ◆ In Store Administration, apply a credit to the order. You should receive a credit confirmation. Is the e-mail formatted as expected?
- ◆ In Store Administration, ship the order. You should receive a shipping confirmation. Is the e-mail formatted as expected?

Troubleshooting

This section provides basic troubleshooting tips.

Troubleshooting Validation Errors

Multiple, non-sensical, errors are reported

Situation: You’ve made a minor change to a template, but the Validator reports many errors, which don’t make any sense.

Solution: Check the code prior to the location of the first error for any tag attributes that are missing a quote (for example, there is a starting quote, but not an ending quote).

Troubleshooting Images

Image is Broken

Situation: You’re viewing a page with either a broken image, or missing image.

Solution: Check for the following:

- ◆ Ensure that you've uploaded the image as the correct type. Images in your header and footer should be uploaded as "Store Design Images", but product, category and manufacturer images should be uploaded as "Shared Images." See *Uploading Images* (page 50) for more information.
- ◆ Check that you're using the correct code to refer to the image.
 - ◆ **Product Thumbnail:** `<ss:image source="$catalog.images[$product.thumbnailImageName]" />`
 - ◆ **Product Image:** `<ss:image source="$product.photoImage" />`
 - ◆ **Manufacturer Image:** `<ss:image source="$catalog.images[$oem.oemLogoName]" />`
 - ◆ **Category Image:** `<ss:image source="$listing.category.photoImage" />`
(assumes this tag is within an `<ss:foreach item="listing" within="$catalog.categoryList()"></ss:foreach>` tag)
 - ◆ **Store Design Image:** `<ss:image source="$store.Images['imagename.ext']">`

Old image still appears after uploading a new one

Situation: You've uploaded a new image that overwrites an existing image used in your store. But when you view the page containing the image, the old image appears.

Solution: Check the following:

- ◆ First, did the image you uploaded have the EXACT same file name as the original? If not, rename the file and upload it again.
- ◆ Next, did you refresh the page after uploading the image? If not, click Refresh. Some browsers may require that you perform a "hard" refresh - please refer to your browser documentation for more information.
- ◆ If you're still having trouble, try clearing your browser's temporary files, close the browser, reopen it and check the page again.

Troubleshooting Pages

Page Appears Broken in Netscape

Situation: You can see a page in Internet Explorer, but not in Netscape.

Solution: Chances are you are missing a "`</td>`" tag somewhere in your code. Review the template to ensure that all `<td>` tags have matching `</td>` tags. If the template is very complex, you might want to export the template and use an HTML Validator tool to review it.

Tags are Not Resolving on a Page

Situation: When viewing a page in your storefront, you see ProStores tags (`<ss:value source="$product.price" />` instead of \$30.00).

Solution: Check for the following:

- ◆ Check the tag syntax to ensure that it is spelled correctly. See the online tag help or the Designer's Reference for more information.
- ◆ Ensure that the Business Object member may be used on the page. See the online template help or the Designer's Reference for more information.
- ◆ If you have downgraded from one version of ProStores to another, you may still have Business Object members in your storefront that only work in the edition you are no longer using. Refer to the Designer's Reference to ensure that the tag may be used by your store.
- ◆ If you are editing your templates in an external editor, check the source code to see that it is not "escaping" the tags - for example, changing "<ss:link/>" to "<ss:link/>".
- ◆ Did you type the tag directly in Design Mode? You can tell if you did by viewing the **Source Mode** tab, and seeing if the tag's opening and closing symbols have been escaped - for example, changing "<ss:link/>" to "<ss:link/>". If you did, delete the tag, then add it using the Tag element.

The Spacing at the Bottom of the Catalog List is Too Wide in Netscape

Situation: You selected the "3 column" catalog list layout in Store Design, and sometimes, the spacing at the bottom of the Catalog List template is too wide.

Solution: This is because the 3 column layout requires that your search results "Max Per Page" setting be a multiple of 3. To change this, go to Store Manager | General Preferences.

Object is not resolving on the storefront

Situation: No errors are received when validating, but the object is not resolving on the storefront.

Solution: Check the object on the template to ensure that a "\$" is preceding it.

Troubleshooting Search Results

Products Appear Multiple Times in Search Results

Situation: When searching, one product will appear multiple times in the search results, in different categories.

This occurs when the following conditions are met:

- ◆ A product is assigned to multiple categories.
- ◆ The Catalog List template is sorted by category.
- ◆ The customer performs a search that returns multiple categories that contain this product.

Solution: This behavior is by design.

You can avoid this behavior by editing the Catalog List template, and sorting it by product instead. Doing so will display all products alphabetically, with no category headings.

However, this may not be an appropriate solution if your customers are searching for products using the **Category** search option. In this situation, it is recommended that you create custom Category List templates that are sorted by category. See *Import and Replace Page Template* (page 33) for more information.

Appendix A: Glossary

alt attributes

An attribute and value added to an `` or `<ss:image>` tag that describes the image (and can be viewed by a visitor when they “mouse over” the image). For example, `<ss:image source="$product.photoImage" alt="$product.description"/>` will display the product description when the visitor mouses over the product’s photo image.

attribute

A property or characteristic of a tag. Most tags require at least one attribute. For example, in `<ss:image source="$product.photoImage"/>`, *source* is an attribute of the `<ss:image/>` tag.

background color

A color that fills an area (either the entire web page area, or just a portion of it).

background pattern

An image (GIF or JPEG) that fills an area (either the entire web page area, or just a portion of it).

baseline templates

Saves a copy of all the templates in a store version.

browser

An application that allows you to look at and interact with the World Wide Web. Examples of browsers include Microsoft Internet Explorer, Netscape Navigator, and Mozilla FireFox.

cart

Contains any items the shopper has expressed interest in purchasing.

catalog

Contains all product information, and can be searched by the shopper.

cookie

Information that a web site puts on a user’s local computer so that it can remember something about them at a later time. Typically, a cookie records preferences when using a particular site.

css (Cascading Style Sheets)

Provides the formatting definitions (such as font, colors, etc) for a web page.

customer

A person who has shopped at your store.

domain name

Locates an organization or other entity on the Internet. For example, <http://www.mystore.com/>.

e-commerce

The buying and selling of goods and services on the Internet.

encryption

The conversion of data into a form (cipher text), that cannot be easily understood by unauthorized people.

ftp (File Transfer Protocol)

A method of exchanging files over the internet. Often used to upload images and files from a local machine to a storefront.

gif

One of the two most common file formats for graphic images on the World Wide Web. The other is the JPEG.

handlebars

Small boxes that surround a selected object.

hexadecimal colors

A system of identifying colors that uses a six-character combination of numbers and letters. For example, white is represented by “#FFFFFF”, black is represented by “#000000” and silver is represented by “#c0c0c0”.

html (Hypertext Markup Language)

The set of code and symbols used to tell a browser how to display a page on the World Wide Web.

jpeg

One of the two most common file formats for graphic images on the World Wide Web. The other is the GIF.

merchant

The store owner.

merchant account

An account that allows the store to accept credit cards as a method of payment.

meta tag

An HTML tag that describes some aspects of the contents of a Web page. Typically used by search engines to help potential visitors find your site.

multimedia

The combination of text, sound, and/or motion video. For example, music and Flash files are considered multimedia.

object expression

An SSML object expression defines where data is coming from, or the data to which input will be posted. Object expressions always start with the “\$” character. For example, in `<ss:image source="$product.photoImage" />`, `$product.photoImage` is the object expression which points to a product’s photo image.

pixel

The basic unit of programmable color on a computer display or in a computer image. The physical size of a pixel depends on how the resolution has been set for the display screen.

publish templates

The act of applying any changes to a template to your “live” storefront.

reset templates

The act of reverting your templates to the last saved baseline.

shopping cart

See “cart”.

spider

A program that visits Web sites and reads their pages and other information in order to create entries for a search engine index.

splash page

A page that greets your customers before they go to your main storefront. Splash pages can also serve as the page that “spiders” can catalog (although not all spiders will catalog splash pages).

ssl (Secure Sockets Layer)

SSL is a Secure Sockets Layer that provides security in transferring data over the Internet, forming a layer of encryption between the server and client. This is particularly important in a commerce situation, as the shopper is providing merchants with their personal information and often with their credit card number.

ssml (ProStores Markup Language)

A combination of tags and object expressions that allow you to work with information from a store.

url

The “address” of a web page. For example, both `http://www.mystore.com/` and `http://www.stores.com/mystore/` are URLs.

visitor

Someone who has browsed your store, but has not necessarily purchased anything.

wysiwig

“What you see is what you get”. WYSIWIG editing allows you to edit web pages while having an idea of how they’ll look. Source code editing only allows you to see and work with the code behind a web page. ProStores Store Design offers a WYSIWIG editing mode

Appendix B: Template Reference

This appendix provides information on each of the templates provided by ProStores. Additional user-defined templates can be added to a store version using the Store Design in the ProStores Store Administration.

When working with SSML, templates can be linked to using the TemplateSet Business Object. System defined templates can be linked to using `<ss:link source="$templateSet.cart"/>`. Custom templates can be linked to using `<ss:link source="$templateSet.Templates['templateshortname']"/>`.

About Us All

General information about your store.

Short Name : about

Link to using: `<ss:link source="$templateSet.aboutUs"/>`

Anonymous Customer Checkout All

Short Name : checkoutanon

Link to using: Cannot link directly to this template

Auction Outcome Email All

E-mail message that notifies the customer when they have won an eBay listing, and includes a link that they must click to begin the checkout process.

Short Name : auctionwon-email

Link to using: Cannot link directly to this template

Billing Options All

Short Name : billing

Link to using: Cannot link directly to this template

Cart

All

Customer shopping cart.

If you want the customer to be able to modify the contents of their cart, you must include the `<ss:form shortcut="CartUpdate"/>` tag somewhere on the detail line. This is the default. If you do not, the `<ss:edit source="$detail.quantity" name="qty"/>` tag will not allow text to be entered.

Short Name : cart

Link to using: `<ss:link source="$templateSet.cart"/>`

Cart Add

All

Quantity, size, color and other product purchase options, displayed on the Catalog Detail template (at the location of the `<ss:include template="cartadd"/>` tag).

Short Name : cartadd

Link to using: Cannot link directly to this template

Cart Save

Enterprise

Saves the selected shopping cart.

Short Name : cartsave

Link to using: `<ss:link source="$templateSet.cartSave"/>`

Cart Select

Enterprise

Displays the customer's saved shopping carts.

Short Name : cartselect

Link to using: `<ss:link source="$templateSet.cartSelect"/>`

Catalog

All

Product search page.

The default number of products to be returned is set on the Store Manager | General Preferences page.

Short Name : catalog

Link to using: <ss:link source="\$templateSet.catalog"/>

Catalog - Members Only

Enterprise

Product search page for logged on members.

Short Name : cataloga

Link to using: <ss:link source="\$templateSet.searchMembersOnly"/>

Catalog Detail

All

Detailed product information.

Short Name : catalogdetail

Link to using: <ss:link source="\$templateSet.productDetail"/>

Catalog Detail - Members Only

Enterprise

Detailed product information for logged on members.

The members only Catalog Detail template is identical in function to the standard Catalog Detail template , except that the customer may be required to provide a valid log on ID.

Short Name : catalogdetaila

Link to using: <ss:link source="\$templateSet.productDetail"/>

Catalog Detail - Related Products

Enterprise

A list of products that have been attached to this product using the Related Products feature. This template is displayed within the Catalog Detail template (at the location of the <ss:include template="catalogrelated"/> tag).

Short Name : catalogrelated

Link to using: Cannot link directly to this template

Catalog List

All

Search results list.

If you are using either the 2 or 3 column layout, you may need to change your Max Per Page setting on the Store Manager | General Preferences page. The Max Per Page setting should be a multiple of the number of columns you are using in the layout (for example, if you are using the 3 column layout, the setting should be 6, 9, 12, etc.). In addition, there are instructions in the template source code itself to assist you in changing the number of columns in the layout.

Short Name : cataloglist

Link to using: <ss:link source="\$templateSet.searchResults"/>

Catalog List

Enterprise

Search results list for logged on members.

If you are using either the 2 or 3 column layout, you may need to change your Max Per Page setting on the Store Manager | General Preferences page. The Max Per Page setting should be a multiple of the number of columns you are using in the layout (for example, if you are using the 3 column layout, the setting should be 6, 9, 12, etc.). In addition, there are instructions in the template source code itself to assist you in changing the number of columns in the layout.

Short Name : cataloglista

Link to using: <ss:link source="\$templateSet.searchResults"/>

Catalog List - Not Found

All

Message displayed if no products meet the customer's search criteria.

Short Name : cataloglisteof

Link to using: Cannot link directly to this template

Checkout

All

Starting point of the checkout process. From this page, customers can select a shipping option, existing customers can log in, and new customers can register.

In order to checkout as a returning customer, they must provide their e-mail address and password.

To allow the customer to indicate that a purchase is for resale, add the following tags to the Checkout and Checkout New Customer templates: `<ss:checkbox source="resale"/><ss:edit source="$customer.resale"/>`.

Short Name : checkoutpro

Link to using: `<ss:link source="$templateSet.checkout"/>`

Checkout Deadend

All

Message displayed if the customer's credit card does not meet your payment requirements.

Specifically, it will be displayed if your store only accepts credit orders from customers with a payment address from the store's home country; the customer has entered a different country than the store's home country; and there are no other payment options available for the customer (such as check, COD, etc.)

Short Name : checkout-deadend

Link to using: Cannot link directly to this template

Checkout New Customers

All

Gathers new customer information.

To allow the customer to indicate that a purchase is for resale, add the following tags to the Checkout and Checkout New Customer templates: `<ss:checkbox source="resale"/><ss:edit source="$customer.resale"/>`.

Short Name : checkout

Link to using: `<ss:link source="$templateSet.checkoutRegistration"/>`

Closed

All

Message displayed if the store is closed.

Short Name : closed

Link to using: Cannot link directly to this template

County Tax

All

Allows customers to select the county in which they live, if the customer's zip/postal code spans more than one county (and therefore, more than one county tax zone).

Short Name : countytax

Link to using: Cannot link directly to this template

Credit

All

Credit detail.

Short Name : credit

Link to using: Cannot link directly to this template

Credit - Email

Enterprise

Confirms a customer credit by e-mail.

Short Name : credit-email

Link to using: Cannot link directly to this template

Customer History

All

Customer transaction history.

Short Name : history

Link to using: <ss:link source="\$templateSet.customerHistory"/>

Customer Properties

All

Displays current customer information, and allows the customer to update that information.

Short Name : customer

Link to using: <ss:link source="\$templateSet.customerProfile"/>

Customer Service

All

Customer service information and links.

Use this template to provide a link to the customer's order history and lost password pages. You can also allow a customer to look up an order's status via the tracking number. Remember to display the tracking number in the Order Confirm template.

Short Name : service

Link to using: <ss:link source="\$templateSet.customerService"/>

Customer Sign In

All

Log in form for previous customers who wish to view their account history.

All fields are required.

Short Name : signin

Link to using: Cannot link directly to this template

Customer Subscription

Enterprise

Information regarding the selected subscription.

Short Name : subscription

Link to using: <ss:link source="\$templateSet.subscribe"/>

Download List

Enterprise

Displays the list of items that are available for download.

Short Name : downloadlist

Link to using: <ss:link source="\$templateSet.downloadList"/>

Email A Friend

All

Allows a visitor to e-mail a link to your store's home or product detail page to a friend.

Short Name : emailfriend

Link to using: <ss:link source="\$templateSet.emailAFriend"/>

Email A Friend Email Content

All

The e-mail containing a link to your store sent by a visitor to a friend.

Short Name : emailfriend-email

Link to using: Cannot link directly to this template

Email A Friend Result

All

Displays the information that was entered on the Email a Friend template.

Short Name : emailfriendresult

Link to using: Cannot link directly to this template

Footer

All

Defines the overall page layout. This template works in conjunction with the Header template.

You can achieve many page layouts with the Header and Footer templates. These are used on every page of the store. All other templates are displayed within the area defined by these templates.

In addition, this template will *not* display on any template that begins with an <HTML> tag. This allows you to create custom designs on individual templates.

Short Name : footer

Link to using: Cannot link directly to this template

Frequently Asked Questions

All

Answers to frequently asked questions.

Use this template to improve the level of customer service to your customers. If your business is getting the same question repeatedly from your customers, you should add it to this template.

If this template does not appear, you can add it using the Site Connector.

Short Name : faq

Link to using: <ss:link source="\$templateSet.faq"/>

Generated Account Password - Email

All

E-mail that provides a customer with the password that the ProStores automatically generated for them.

This template will only be sent if a customer tries to log on or check out using a customer record that does not have a password. The ProStores will generate the password and send it to the customer.

Short Name : genpass-email

Link to using: Cannot link directly to this template

Header

All

Defines the overall page layout. This template works in conjunction with the Footer template.

You can achieve many page layouts with the Header and Footer templates. These are used on every page of the store. All other templates are displayed within the area defined by these templates.

In addition, this template will *not* display on any template that begins with an <HTML> tag. This allows you to create custom designs on individual templates.

Short Name : header

Link to using: Cannot link directly to this template

Invoice

All

Order invoice.

If you have upgraded from a prior version of the ProStores, and are using a shipper other than/in addition to the ProStores, you will need to replace the line containing the shipping information with the <ss:include template="selectshipping"/> tag.

This template can also be viewed in Store Administration.

If you use the UPS shipper, the ProStores displays the UPS Trademark and accompanying disclaimer on each invoice and shipping related screen viewed by your customers. These

elements are permanent and may not be modified or deleted. If these fields do not appear, add the `<ss:include macro="ShipperCopyright" />` to this template.

Short Name : invoice

Link to using: Cannot link directly to this template

Invoice - Email

All

Order invoice that can be e-mailed to a customer.

If you use the UPS shipper, the ProStores displays the UPS Trademark and accompanying disclaimer on each invoice and shipping related screen viewed by your customers. These elements are permanent and may not be modified or deleted. If these fields do not appear, add the `<ss:include macro="ShipperCopyright" />` to this template.

Short Name : invoice-email

Link to using: Cannot link directly to this template

Invoice - Print

All

Printable order invoice (does not contain the header or footer).

This template is used only in Store Administration to print an invoice.

If you use the UPS shipper, the ProStores displays the UPS Trademark and accompanying disclaimer on each invoice and shipping related screen viewed by your customers. These elements are permanent and may not be modified or deleted. If these fields do not appear, add the `<ss:include macro="ShipperCopyright" />` to this template.

Short Name : invoice-print

Link to using: Cannot link directly to this template

Lost Password

All

Verifies customer information so that password information may be e-mailed to them.

If the first and last name fields are present on this template, they will be required entry fields for the customer. If you delete the fields, the password will be sent to the customer based on the e-mail address alone.

Short Name : lostpassword

Link to using: <ss:link source="\$templateSet.lostPassword"/>

Lost Password - Email

All

E-mail providing the user name and password to a customer who has lost their information.

Short Name : lostpass-email

Link to using: Cannot link directly to this template

Low Inventory - Email

Enterprise

E-mail that will be sent to the Store Buyer if Buyer Notifications are enabled, and inventory for a product is below the product's threshold.

Short Name : lowinv-email

Link to using: Cannot link directly to this template

Member Login

Enterprise

Login form for registered members.

A valid e-mail and password is required. Customers obtain passwords by either purchasing a product from the store, or by registering with the store.

All fields are required.

Short Name : memberlogin

Link to using: <ss:link source="\$templateSet.logon"/>

Member Login Error

All

Message displayed if member login information is incorrect.

Short Name : memberloginerror

Link to using: Cannot link directly to this template

Message

All

Miscellaneous messages that are presented to the customer.

These include when a required form field is not entered, when an order is cancelled, and when any other error condition occurs.

Short Name : message

Link to using: Cannot link directly to this template

Order Confirm

All

Order confirmation page.

Short Name : orderconfirm

Link to using: Cannot link directly to this template

Order Extensions

Enterprise

Gathers custom product information on a per product basis.

If this template is defined in a product's profile, it will be shown once for each product purchased. This includes once for each different product multiplied by the quantity ordered.

This template can also be viewed in Store Administration.

Short Name : orderext

Link to using: Cannot link directly to this template

Packing Slip

All

Packing slip for an order, displayed without the header or footer.

This template is used only in Store Administration - it is not visible in the storefront.

Short Name : packing

Link to using: Cannot link directly to this template

PayPal Purchase Token

All

Begins the PayPal payment process. This template is displayed within the Order Confirm template (at the location of the `<ss:include template="paypal-purchtoken"/>` tag).

This template conforms to Pay Pal's payment processing requirements. Changes should not be made unless instructed otherwise by Pay Pal, or your Site Administrator.

Short Name : paypal-purchtoken

Link to using: Cannot link directly to this template

Policies

All

Store policies, such as shipping policies, returns and credits.

Short Name : policy

Link to using: `<ss:link source="$templateSet.storePolicies"/>`

Privacy Statement

All

Store privacy statement.

Privacy is a very important issue for customers shopping online. Some countries have strict privacy requirements for those business that wish to sell to their citizens. You are strongly encouraged to familiarize yourself with the privacy requirements of the countries in which you will do business.

The text shipped with the ProStores is only intended to serve as sample text.

If this template does not appear, you can add it using the Site Connector.

Short Name : privacy

Link to using: `<ss:link source="$templateSet.privacyPolicy"/>`

Product Authorization

Enterprise

Message displayed if the customer is not authorized to purchase one or more products in their cart.

The authorization check occurs during checkout. Authorization is granted to a customer by selecting the **Authorization** check box in the Product Profile.

Short Name : productauth

Link to using: Cannot link directly to this template

Promotion Alert

All

Notifies the shopper if they are eligible for other promotions that would benefit them.

Short Name : promotion-alert

Link to using: Cannot link directly to this template

Purchase Order Limit

Enterprise

Message displayed if an order exceeds the customer's purchase order limit (defined in their Customer Profile).

You may want to include a link to your Service page or a specific way to contact the person for approving order limits.

Short Name : polimit

Link to using: Cannot link directly to this template

Recurring Bill Error - Email

Enterprise

Notifies staff (using the staff e-mail address), that an error occurred when payment a subscription.

Short Name : recurerror-email

Link to using: Cannot link directly to this template

Recurring Billing - Email

Enterprise

E-mail sent when a subscription is billed to the customer.

Short Name : recurbill-email

Link to using: Cannot link directly to this template

Registration

All

Begins the store registration process.

Short Name : register

Link to using: <ss:link source="\$templateSet.customerRegistration"/>

Registration Confirmation

All

Confirms that the customer has registered with the store.

Short Name : regconfirm

Link to using: Cannot link directly to this template

Sales Affiliate

Sales Manager

Registers a sales affiliate.

Sales Affiliates are other businesses, individuals or organizations to whom you agree to pay a small percentage of the item's sale price if they refer a customer to your store.

Short Name : affiliate

Link to using: <ss:link source="\$templateSet.affiliate"/>

Sales Affiliate Confirm

Sales Manager

Confirms the registration of a sales affiliate.

Sales Affiliates are other businesses, individuals or organizations to whom you agree to pay a small percentage of the item's sale price if they refer a customer to your store.

Short Name : affilconfirm

Link to using: Cannot link directly to this template

Sales Affiliate Email

Sales Manager

Confirms the registration of a sales affiliate by e-mail.

Sales Affiliates are other businesses, individuals or organizations to whom you agree to pay a small percentage of the item's sale price if they refer a customer to your store.

Short Name : affiliate-email

Link to using: Cannot link directly to this template

Sales Lead

Sales Manager

Gathers sales lead information from a customer.

Sales Leads are generated when a Product's Type field is set to "Lead".

Short Name : lead

Link to using: Cannot link directly to this template

Sales Lead Confirmation

Sales Manager

Confirms the sales lead.

Sales Leads are generated when a Product's Type field is set to "Lead".

Short Name : leadconfirm

Link to using: Cannot link directly to this template

Select Shipping

All

Shipping quotes list, displayed within the Invoice template.

Short Name : selectshipping

Link to using: Cannot link directly to this template

Ship To

All

Collects shipping information if different from the customer's payment address.

This template is used if the Gift option is NOT selected on the Store Manager | Promotions page. Otherwise, the Ship To Gift template is used.

Short Name : shipto

Link to using: Cannot link directly to this template

Ship To Gift

Enterprise

Collects shipping and gift wrap information, if gift wrap is enabled for the store.

Gift Shipments must be enabled on the Store Manager | Promotions page for this template to appear.

The `<ss:value source="$invoice.message"/>` tag can be shown on the Invoice and Packing Slip.

Short Name : shiptogift

Link to using: Cannot link directly to this template

Shipping - Email

Enterprise

E-mail sent when an order is shipped.

Short Name : shipping-email

Link to using: Cannot link directly to this template

Store Locator

All

Information regarding the physical locations of your stores.

If this template does not appear, you can add it using the Site Connector.

Short Name : locator

Link to using: `<ss:link source="$templateSet.storeLocations"/>`

Storefront

All

Store home page.

Short Name : storefront

Link to using: `<ss:link source="$templateSet.homePage"/>`

Subscription

Enterprise

Starting page in the store subscription process.

This template appears when the customer clicks the `<ss:link source="$templateSet.subscribe">Become a Member</ss:link>` tag in a store that has a subscription product based members only area.

Short Name : subscribe

Link to using: Cannot link directly to this template

Subscription Billing Option

Enterprise

Short Name : recurbilling

Link to using: Cannot link directly to this template

Subscription Cancellation - Email

Enterprise

E-mail sent to customers when a subscription is cancelled.

Short Name : subscancel-email

Link to using: Cannot link directly to this template

Subscription Renewal - Email

Enterprise

E-mail sent to customers when a subscription is renewed.

Short Name : subscreenew-email

Link to using: Cannot link directly to this template

Supplier EDI

Supplier Connect

Notifies suppliers via download of orders that have been placed for their products.

This template contains Header (“H”) and Detail (“D”) records, which are required for proper downloading. The header record contains the customer’s number, invoice number, and order date. The detail records contain the customer’s number, invoice number, orderdate, product SKU, and product quantity. The resulting file is comma delimited with each string field enclosed in double quotes.

Short Name : supplier1

Link to using: Cannot link directly to this template

Supplier EDI 850

Supplier Connect

Notifies suppliers via ANSI ASC X12 download of orders that have been placed for their products.

The template uses the ANSI ASC X12 (American National Standards Institute Accredited Standards Committee X12, form 850 (Purchase Order) format. Changes should be made carefully, making sure that any changes conform to those standards.

By default, this template supports one “segment”. If you modify the template to support more than one segment, you will need to modify the parameter in the "`<ss:eval source=\"$detailCount.set($detailCount.multiply(1))\" />`" tag to reflect the number of segments.

Short Name : supplier.x12.850

Link to using: Cannot link directly to this template

Supplier Email

Supplier Connect

Notifies suppliers via e-mail of orders that have been placed for their products.

Short Name : supplier3

Link to using: Cannot link directly to this template

Supplier Fax

Supplier Connect

Notifies suppliers by fax of orders that have been placed for their products.

Short Name : supplier2

Link to using: Cannot link directly to this template

Tax Region Select

All

Allows the customer to preview product pricing inclusive of applicable sales tax, and can be displayed within the Cart, Catalog Detail or Catalog List templates, at the location of the "`<ss:include source="taxregionselect" />`" tag.

Your store must be configured to show prices inclusive of sales tax on the Sales Tax Preferences page in Store Administration before you can use this template.

Short Name : taxregionselect

Link to using: Cannot link directly to this template

Tracking

All

Package tracking information. Only available when using a shipper other than/in addition to the ProStores.

If you use the UPS shipper, the ProStores displays the UPS Trademark and accompanying disclaimer on each invoice and shipping related screen viewed by your customers. These elements are permanent and may not be modified or deleted. If these fields do not appear, add the If these fields do not appear, add the `<ss:include macro="ShipperCopyright"/>` to this template.

Short Name : tracking

Link to using: `<ss:link source="$templateSet.shipmentTracking"/>`

User Defined

All

User Defined templates are those templates that are imported/added with a type of "User Defined".

Short Name : userdefined

Link to using: Cannot link directly to this template

WorldPay Callback Processing Error

All

Message displayed when an error occurs for an order paid for with WorldPay.

This template will only be used if you have set up a Callback URL at your WorldPay account. If you have not done so, then the default page from WorldPay will be displayed instead. Tags that begin `<WP*>` are WorldPay tags and should not be edited. They may, however, be moved. In addition, WorldPay requires certain HTML restrictions as well. Instructions regarding these tags and restrictions are available from WorldPay.

Short Name : wpay-error

Link to using: Cannot link directly to this template

WorldPay Payment Cancellation

All

Message displayed when an order paid for with WorldPay has been cancelled.

Tags that begin <WP*> are WorldPay tags and should not be edited. They may, however, be moved. In addition, WorldPay requires certain HTML restrictions as well. Instructions regarding these tags and restrictions are available from WorldPay. This template will only be used if you have set up a Callback URL at your WorldPay account. If you have not done so, then the default page from WorldPay will be displayed instead.

Short Name : wpay-cancel

Link to using: Cannot link directly to this template

WorldPay Payment Confirmation

All

Message displayed when an order paid for with WorldPay has been confirmed.

Tags that begin <WP*> are WorldPay tags and should not be edited. They may, however, be moved. In addition, WorldPay requires certain HTML restrictions as well. Instructions regarding these tags and restrictions are available from WorldPay. This template will only be used if you have set up a Callback URL at your WorldPay account. If you have not done so, then the default page from WorldPay will be displayed instead.

Short Name : wpay-confirm

Link to using: Cannot link directly to this template

WorldPay Purchase Token

All

Begins the WorldPay payment process. This template is displayed within the Order Confirm template.

This template conforms to WorldPay's payment processing requirements changes should not be made unless instructed otherwise by WorldPay, or your Site Administrator.

Short Name : wpay-purchase

Link to using: Cannot link directly to this template

Appendix C: Tag Reference

This reference describes the SSML Tags available to use in the storefront, with samples.

The following is an example of how SSML tags are constructed:

<code><ss:edit</code>	<code>source="\$customer.FirstName"</code>	<code>size="20"/></code>			
(1)	(2)	(4)	(3)	(4)	(6)

<code><ss:link</code>	<code>source="\$templateSet.aboutUs"></code>	<code>About Us</code>	<code></ss:link></code>		
(1)	(2)	(4)	(3)	(5)	(6)

- 1: `<ss:/>` defines this as a ProStores tag.
- 2: The tag type. Generally indicates what this tag will resolve to on the storefront.
- 3: The Object Expression. Defines where the data is coming from, or the data to which input will be posted. Object expressions always start with the "\$" character.
- 4: Tag attributes that further define the tag.
- 5: The content to appear between the opening and closing tags. Not all tags have content.
- 6: The closing slash or tag.

<ss:break>

Allows you to "break" out of an `<ss:foreach>` loop.

Examples

To display only the first 10 top-level categories on the header, use:

<pre><ss:foreach item="category" within="\$catalog.categoriesToDepth(1)"> <ss:link source="\$category"/>
 <ss:if test="\$counter.currentRow==10"> <ss:break/> </ss:if> </ss:foreach></pre>

<ss:button>

Creates a button for a form (either as a standard form button, or as an image button).

Examples

To add a Submit button to a form, use:

```
<ss:button type="submit" value="Submit" />
```

...which resolves to:

```
<input type="submit" value="Submit">
```

To add a Submit button to a form that appears as an image, use:

```
<ss:button source="$templateSet.images['go.gif']" alt="Search" border="0" />
```

...which resolves to:

```
<input type='image' src="/.../go.gif" border="0" alt="Search">
```

Attributes

Supports HTML attributes

source

Use this attribute to point to an image object expression when creating an image button.

type

The type of button. Only use this attribute to override the type attribute that ProStores generates.

<ss:checkbox>

Creates a checkbox for a form.

Examples

To allow the shopper to add the gift wrap service to their order, use:

```
<ss:checkbox source="$invoice.giftWrapEnabled" />
```

...which resolves to:

```
<input type="checkbox" name="$invoice.giftWrapEnabled" value="1" />
```

To allow the shopper to designate the purchase for resale (a valid resale certificate must be on file for the shopper), use:

```
<ss:checkbox source="resale" /> This purchase is for resale.
```

...which resolves to:

```
<input type="checkbox" name="resale" /> This purchase is for resale.
```

Attributes

Supports HTML attributes

checked

Use this attribute to select the check box by default.

name

The form control name. Should only be used if the name is different than the source attribute.

shortcut

Shortcut attributes allow you to replace a large, complex piece of code with one word. All formatting and text are included within the shortcut. If you want to control formatting or text, replace the shortcut with the corresponding code (for assistance, try viewing the HTML source of the storefront page).

Valid Values

Value	Description
ChangeBilling	Creates a check box that allows the shopper to indicate that they wish to change their payment information.
ChangeShippingAddress	Creates a check box that allows the shopper to indicate they wish to ship the order to a different address.

source

The object expression in which the check box value will be stored.

value

The value to be applied if the check box is selected.

<ss:choose>

The <ss:choose> tag displays content from one among a number of possible alternatives. It consists of a sequence of <ss:when> elements, followed by an optional <ss:otherwise> element (which will be applied when *none* of the <ss:when/> conditions are met).

Examples

To apply special formatting to the “shipped” status on an invoice, use:

```
<ss:choose>
  <ss:when test="$detail.status == 'Shipped'">
    <b>This order has been shipped</b>
  </ss:when>
  <ss:otherwise>
    <ss:value source="$detail.status"/>
  </ss:otherwise>
</ss:choose>
```

...which resolves to:

```
<b>This order has been shipped</b>
```

...if shipped. If the order has not been shipped, it displays the current order status (without bold formatting).

To display a list of categories and subcategories, use:

```
<ss:choose>
  <ss:when test="$category.TreeLevel == '2'">
    --<ss:value source="$category.name" /><br>
  </ss:when>
  <ss:when test="$category.TreeLevel == '3'">
    ----<ss:value source="$category.name" /><br>
  </ss:when>
  <ss:otherwise>
    <ss:value source="$category.name" /><br>
  </ss:otherwise>
</ss:choose>
```

...which resolves to:

```
Root Category 1<br>
--Subcategory 1<br>
--Subcategory 2<br>
----Subcategory 3<br>
Root Category 2<br>
...
```

This code would appear within an `<ss:foreach/>` tag that iterates through the list of categories. First, this code checks the level of the current category. If it is at level '2', it displays the category name with two dashes preceding it. If it is at level '3', it displays the category name with four dashes preceding it. Otherwise, it just displays the category name. Category levels correspond to the level of nesting. For example, a top-level category has a level of 1, any subcategories assigned to it have a level of 2, any subcategories assigned to the level 2 subcategories have a level of 3, and so on.

<ss:comment>

Contains comments that will be removed when the template is resolved. Standard HTML comment tags (`<-- -->`) *are not* removed when the template is displayed and are therefore accessible in the page source. If you wish to “hide” any content or code containing ProStores tags, use this tag instead of HTML comments.

Examples

To display a note that cannot be seen when viewing page source, use:

```
<ss:comment>Note Text Here</ss:comment>
```

The code surrounded by the tags will not appear.

<ss:directive>

Communicates specific information to ProStores.

Examples

To display custom keywords in the page's `<meta>` tag, use:

```
On the Header template:
```



```
<meta name="Keywords" content="$page.getDirective('page.meta.keywords') ">
```

On the storefront template:

```
<ss:directive name="page.meta.keywords" value="custom, keywords"/>
```

The `<meta>` tag should be placed in the header, while the `<ss:directive/>` tag should be placed on the body template.

Attributes

name

The item to which information is being communicated

value

The information to be communicated

<ss:edit>

Creates a text input field for a form. If an HTML size attribute is not declared, the default size will be “10”. ProStores adds a “maxlength” attribute, which defines the maximum number of characters allowed for this field.

Examples

To add a field to collect the customer’s first name, use:

```
<ss:edit source="$customer.firstName" size="20">
```

...which resolves to:

```
<input type="text" name="$customer.firstName" value='' maxlength="20" size="20">
```

Attributes

Supports HTML attributes

default

The default value to display in the edit field.

format

Defines the formatting characteristics for this tag when resolving the “source” attribute.

Valid Values

Value	Description
date.short	Date
date.medium	Date
date.full	Date
date.long	Date

Value	Description
time.short	Time
time.medium	Time
time.full	Time
time.long	Time
time24	Time
day	Day
month	Month
year	Year
percent	Percentage
%	Percentage
%N	Percentage, with the number of decimal places specified by 'N'
truefalse	True or False
yesno	Yes or No
none	No formatting
money	Currency

maxlength

The maximum number of characters that may be entered. If this attribute is not specified, the value defined by ProStores will be used.

name

The control name. Should only be used if the name is different than the source attribute.

source (Required)

The object expression in which the edit value will be stored.

value

The default text to appear in the field.

<ss:else>

See the <ss:if> tag for information.

<ss:emailheader>

Contains text to be used in an e-mail message header (such as the subject line).

Examples

To define a title to appear on an e-mail's subject line, use:

```
<ss:emailheader>
  Insert Title Text Here
</ss:emailheader>
```

...which resolves to:

```
Insert Title Text Here
```

... in the subject line of the e-mail message.

Attributes

type

Indicates if this is a text or HTML e-mail message.

Valid Values

Value	Description
text/plain	Indicates that this is a standard text e-mail message.
text/html	Indicates that this is an HTML format e-mail message. Not all e-mail clients support HTML e-mail. To provide both a text and HTML version of the e-mail, enter the text version at the very beginning, between <!-- and --> tags.

<ss:eval>

Executes the value in the source attribute, but does not display it.

Examples

To assign a value to the math object, use:

```
<ss:eval source="$math.set(5)"/>
```

The value is set to '5'.

Attributes

source (Required)

The object expression that is to be executed

<ss:font>

Creates HTML font tags that use the values defined in Store Design to format the storefront templates.

Examples

To apply all the heading font attributes, use:

```
<ss:font source="$font.heading">
```

...which resolves to:

```
<font face="arial,helvetica,sans-serif" color="black" size="4">
```

The tag resolves to the selections made using the Fonts and Colors page in Store Design.

To mix the font settings with custom font attributes, use:

```
<ss:font face="$font.paragraph" size="$font.paragraph" color="red">
```

...which resolves to:

```
<font face="arial,helvetica,sans-serif" size="2" color="red">
```

...assuming “arial,helvetica,sans-serif” was the option selected in the Fonts and Colors page in Store Design.

Attributes

Supports HTML attributes

source

The object expression containing the font property.

face

The object expression containing the font face property.

color

The object expression containing the font color property.

size

The object expression containing the font size property.

<ss:foreach>

Iterates over the items in a collection. For example, use this tag (in conjunction with others) to display a list of products. Or, you could use this tag to create a set of links to each of your product categories. In addition, this tag supports the <ss:sortby> tag on catalog list templates, which specifies the sort order of the list.

Examples

To display a list of products, with no sorting, use:

```
<ss:foreach item="listing" within="$catalog.results">  
  <ss:foreach item="product" within="$listing.products">
```

```

        <p><ss:value source="$product.name" /></p>
    </ss:foreach>
</ss:foreach>

```

...which resolves to:

```

<p>Product 1</p>
<p>Product 2</p>
<p>...</p>

```

The first <ss:foreach/> creates the list of search results. Then, the second <ss:foreach> displays that list by product.

To display a list of products, grouped by category, use:

```

<ss:foreach item="listing" within="$catalog.results">
  <ss:sortby item="category" direction="asc" />

  <h1><ss:value source="$category.name" /></h1>

  <ss:foreach item="product" within="$listing.products">
    <ss:sortby item="product" direction="asc" />

    <p><ss:value source="$product.name" /></p>

  </ss:foreach>
</ss:foreach>

```

...which resolves to:

```

<h1>Category A Name</h1>
  <p>Product 1</p>
  <p>Product 2</p>
  <p>...</p>
<h1>Category B Name</h1>
  <p>Product 1</p>
  <p>Product 2</p>
  <p>...</p>
<h1>...</h1>

```

In this example, the first <ss:foreach> creates the list of products, then sorts them by category name. The next <ss:foreach> creates the list of products within each category, and sorts them by product name.

To display the prices and names for all of a product's attributes formatted as a bulleted list, use:

```

<ul>
  <ss:foreach item="attr" within="$product.attributes">
    <li><ss:value source="$attr.price" />, <ss:value
source="$attr.description" /></li>
  </ss:foreach>
</ul>

```

...which resolves to:

```

<ul>
  <li>$85.105 Full-Size</li>
  <li>$19.95 Miniature</li>
</ul>

```

Attributes

item

Specifies the name to be used when referencing object expressions within the collection.

within

Specifies the collection or iterator from which items will be listed. For example, when creating a list of categories, the *category* is the “item”, and *CategorySearch* is the “collection”

<ss:form>

Creates a complete HTML form, as specified by the *shortcut* attribute.

Examples

To allow the customer to search for an order by entering a tracking number, use:

```
<ss:form shortcut="Tracking"/>
```

...which resolves to:

```
<form name='tracking' method='POST' action='...'>
  Tracking No: <input type='text' name='tracking' value='' size=12 maxlength=12>
    <input type='submit' value='Go!'/>
</form>
```

All formatting for the form is defined by the shortcut.

To create a form that allows users to update the quantity of product in their cart, use:

```
<ss:form shortcut="CartUpdate">
  <ss:edit name="qty" source="$detail.Quantity" size="4"/>
</ss:form>
```

...which resolves to:

```
<form name='cart' method='POST' action='...'>
  <input type='hidden' name="key" value="12/23/02 10:49:39 AM">
  <input type='hidden' name="product_no" value="2">
  <input type="text" name="qty" value="1" name="qty" size="4"/>
</form>
```

Attributes

Supports HTML attributes

shortcut (Required)

Shortcut attributes allow you to replace a large, complex piece of code with one word. All formatting and text are included within the shortcut. If you want to control formatting or text, replace the shortcut with the corresponding code (for assistance, try viewing the HTML source of the storefront page).

Valid Values

Value	Description
CartEnd	Inserts the HTML code that ends a form containing a list of items that can be selected and added to a cart. Must be used with a CartStart shortcut.
CartStart	Inserts the HTML code that begins a form containing a list of

Value	Description
	items that can be selected and added to a cart. Must be used with a CartEnd shortcut.
CartUpdate	Allows the customer to update the quantity of items in their cart.
ChangeSubscriptionPayment	Inserts the button to take the customer to the subscription payment information template.
ChangeSubscriptionShipTo	Inserts the button to take the customer to the subscription shipping information template.
Checkout	Inserts the button to take the customer to the Checkout New Customers template.
CheckoutPro	Inserts the button to take the customer to the Checkout template.
LanguageOverride	Inserts a list from which the customer can choose the language in which they want the store to appear.
Pay	Defines the form that allows the customer to pay an outstanding invoice.
PayPalPurchaseToken	Inserts the button to take the customer to complete a PayPal transaction.
SubmitSubscription	Allows the customer to submit changes to their subscriptions.
Tracking	Inserts a form that allows the customer to search for a specific order.

<ss:hidden>

Creates a hidden field for a form.

Examples

To include the current date in a form, without displaying it to the customer , use:

```
<ss:hidden source="$system.currentDate" name="date" />
```

...which resolves to:

```
<input type="hidden" name="date" value="01/01/2003">
```

Attributes

Supports HTML attributes

name

The control name. Should only be used if the name is different than the source attribute.

format

Defines the formatting characteristics for this tag when resolving the “source” attribute.

Valid Values

Value	Description
date.short	Date
date.medium	Date
date.full	Date
date.long	Date
time.short	Time
time.medium	Time
time.full	Time
time.long	Time
time24	Time
day	Day
month	Month
year	Year
percent	Percentage
%	Percentage
%N	Percentage, with the number of decimal places specified by 'N'
truefalse	True or False
yesno	Yes or No
none	No formatting
money	Currency

source (Required)

The object expression in which a value will be stored.

value

The default value associated with the field.

<ss:if>

Tests a single condition, described in the *test* attribute. If you need to test more than one condition, the <ss:choose> tag may be more appropriate. Also includes the <ss:else/> tag.

Examples

To display a selection list only if a product has colors defined for it, use:

```
<ss:if test="$product.hasColors()">
  Color: <ss:select source="$product.colors" name="color"/>
</ss:if>
```

...which resolves to:

```
Color: <select name="color">
  <option value="Blue">Blue</option>
  <option value="Green">Green</option>
```



```
</select>
```

...if the test condition is met. Otherwise, nothing is displayed.

To display the “Save Cart” link only if the customer is logged in *and* has items in their cart, use:

```
<ss:if test="$customer.isLoggedIn() AND $cart.details">  
  <ss:link source="$templateSet.cartSave">Save Cart</ss:link>  
</ss:if>
```

...which resolves to:

```
<a href='/click/Cart.bok?smode=save'>Save Cart</a>
```

...if the test conditions are both met. Otherwise, nothing will be displayed

To change the formatting of a category, based on its level, use:

```
<ss:if test="$category.treeLevel == '1'">  
  <br><ss:value source="$category.name" /><br>  
<ss:else/>  
  --<ss:value source="$category.name" /><br>  
</ss:if>
```

...which resolves to:

```
Root Category 1<br>
```

...if this is a top level category. Otherwise, it will resolve to “--Subcategory 1
”

To display the items a customer has added to their cart, use:

```
<ss:if test="$cart.details">  
  <ss:foreach item="detail" within="$cart.details">  
    <ss:value source="$detail.name" /><br>  
  </ss:foreach>  
<ss:else/>  
  Empty  
</ss:if>
```

...which resolves to:

```
Cutie Doll<br>  
Fancy Girl Doll<br>  
Dining Room Lithograph<br>
```

...if there are products in the cart. Otherwise, it would display “Empty”.

Attributes

test (Required)

Defines the test condition. Includes an Object Expression, operand, and test value. You can test for multiple conditions by using a single “AND” or “OR” statement. AND/OR must either be in all caps, or use the following symbols: “&&” (AND), “||” (OR). Valid operands are:

- ◆ > - Greater Than
- ◆ < - Less Than
- ◆ >= - Greater Than or Equal To
- ◆ <= - Less Than or Equal To

- ◆ = - Equals
- ◆ == - Equals
- ◆ != - Not Equal To
- ◆ <> - Not Equal To
- ◆ MATCHES - Supports wildcard at the end of the string (for example 'string*').

<ss:image>

Includes an image on the template.

Examples

To add a banner to your header template, use:

```
<ss:image source="$storeVersion.storeLogo" border="0" />
```

...which resolves to:

```

```

To display the product photo image on the Catalog List template, use:

```
<ss:image source="$product.photoImage" border="0" />
```

...which resolves to:

```

```

Attributes

Supports HTML attributes

source

The object expression that contains the image you wish to display.

<ss:include>

Inserts a template or file into the current template (at the location of this tag).

Examples

To display the Related Products template within the Catalog Detail template, use:

```
<ss:include template="catalogrelated"/>
```

The code defined in the Related Products template will appear.

To display a static file on any template, use:

```
<ss:include source="$store.files['filename.ext']" />
```

The code defined in the filename.ext file will appear. This example assumes that you have uploaded filename.ext to your store content area.

To display the product price, conditionally displaying sale prices and saving where applicable, use:

```
<ss:include macro="productprice"/>
```

...which resolves to:

```
Product price, plus any sale information and formatting.
```

While the macro attribute “productprice” will determine if any sale information and formatting needs to be displayed, it will not allow you to apply your own formatting.

Attributes

source

The object expression containing the location of the file you wish to include.

template

The template or file to be included.

Valid Values

Value	Description
catalogrelated	Related Products template.
cartadd	Cart Add template.
paypal-purchtoken	PayPal Purchase Token template.
selectshipping	Select Shipping template.
taxregionsselect	Tax Region Select template.
wpay-purchase	World Pay Purchase Token template.
filename	The file name of any user-defined templates.

macro

Code to be included at the location of this tag.

Valid Values

Value	Description
productattributes	<p>The product attributes, with label and display type. The SSML equivalent of this macro is:</p> <pre><ss:if test="\$product.attributeLists"> <ss:foreach item="productAttrList" within="\$product.attributeLists"> <ss:if test="\$productAttrList.attributeValues"> <ss:if test="\$productAttrList.attributeValues.size > 1"> <ss:if test="\$productAttrList.displayType == DisplayType.RADIO"> <table cellpadding="0" cellspacing="0" border="0"> <ss:foreach item="attributeValue" within="\$productAttrList.attributeValues"> <tr> <td nowrap valign="top"> <ss:if test="\$counter.currentRow==1"> <ss:font source="\$font.paragraph"><ss:value source="\$productAttrList.attributeLabel"/>:</ss:font></pre>

Value	Description
	<pre> </ss:if> </td> <td valign="top"> <ss:radio name="\$system.concat('attributevalue_', \$productAttrList.listNumber)" source="\$attributeValue" checked="\$counter.currentRow==1" /> </td> <td width="100%" valign="middle"> <ss:font source="\$font.paragraph"><ss:value source="\$attributeValue" /></ss:font> </td> </tr> </ss:foreach> </table> <ss:else/> <div style="position:relative;margin-top:5px;margin- bottom:5px;"> <ss:font source="\$font.paragraph"><ss:value source="\$productAttrList.attributeLabel" /></ss:font> <ss:select source="\$productAttrList.attributeValues" name="\$system.concat('attributevalue_', \$productAttrList.listNumber)" /> </div> </ss:if> <ss:else/> <div style="position:relative;margin-top:5px;margin- bottom:5px;"> <ss:font source="\$font.paragraph"><ss:value source="\$productAttrList.attributeLabel" /></ss:font> <ss:value source="\$productAttrList.valuesText" /> <input type="hidden" name="\$system.concat('attributevalue_', \$productAttrList.listNumber)" value="\$productAttrList.valuesText" /> </div> </ss:if> </ss:if> </ss:foreach> </ss:if> </pre>
productprice	<p>The product price, with sale price and savings information, if valid. The SSML equivalent of this macro is:</p> <pre> <ss:foreach item="productprice" within="\$product.finalPriceList"> <ss:choose> <ss:when test="\$counter.totalRows == 1"> <ss:value source="\$productprice" /> </ss:when> <ss:otherwise> <ss:choose> <ss:when test="\$counter.isLast"> <ss:value source="\$productprice" /> </ss:when> <ss:otherwise> <s><ss:value source="\$productprice" /></s> </ss:otherwise> </ss:choose> </ss:otherwise> </ss:choose> &nbsp; </ss:foreach> <ss:foreach item="promotion" within="\$product.activePromotions"> <ss:set name="detailText" value="\$product.promotionDetailText(\$promotion)" /> <ss:if test="\$detailText != null"> (<ss:value source="\$detailText" />) </ss:if> </ss:foreach> </pre>

Value	Description
attributedetailprice	<p>The attribute detail price, with sale price and savings information, if valid. The SSML equivalent of this macro is:</p> <pre> <ss:if test="\$product.hasAttributes"> <ss:foreach item="attribute" within="\$product.attributes"> <ss:foreach item="price" within="\$attribute.finalPriceList"> <ss:choose> <ss:when test="\$counter.isLast"> <ss:value source="\$price"/> </ss:when> <ss:otherwise> <s><ss:value source="\$price"/></s> </ss:otherwise> </ss:choose> </ss:foreach> <ss:foreach item="promotion" within="\$product.activePromotions"> <ss:set name="detailText" value="\$product.promotionDetailText(\$promotion)"/> <ss:if test="\$detailText != null"> (<ss:value source="\$detailText"/>) </ss:if> </ss:foreach> <ss:value source="\$attribute.attributeText"/>
 </ss:foreach> </ss:if> </pre>
attributedetailfull	<p>The attribute detail price, with sale price and savings information, if valid. The SSML equivalent of this macro is:</p> <pre> <ss:if test="\$product.hasAttributes"> <ss:foreach item="attribute" within="\$product.attributes"> <ss:foreach item="price" within="\$attribute.finalPriceList"> <ss:choose> <ss:when test="\$counter.isLast"> <ss:value source="\$price"/> </ss:when> <ss:otherwise> <s><ss:value source="\$price"/></s> </ss:otherwise> </ss:choose> &nbsp; </ss:foreach> <ss:foreach item="promotion" within="\$product.activePromotions"> <ss:set name="detailText" value="\$product.promotionDetailText(\$promotion)"/> <ss:if test="\$detailText != null"> (<ss:value source="\$detailText"/>) </ss:if> </ss:foreach> <ss:value source="\$attribute.attributeText"/> <ss:if test="\$store.inventoryEnabled"> <ss:if test="\$attribute.quantity > 0"> &nbsp;<ss:value source="\$attribute.quantity"/> in stock <ss:else/> <ss:if test="\$store.backordersnotallowed"> &nbsp;0 in stock <ss:else/> &nbsp;<BACKORDERD </ss:if> </ss:if> </ss:if>
 </ss:foreach> </ss:if> </pre>

Value	Description
productpricex	See AttributeDetailFull for information regarding this macro.
ShipperCopyright	Information and formatting required by the shipper (such as UPS).

<ss:link>

Creates a link to a template or static page.

Examples

To create a link to the About Us template, use:

```
<ss:link source="$templateSet.aboutUs" />
```

...which resolves to:

```
<a href="../../../page.bok?template=about">About Us</a>
```

To create a link to the Members Only catalog, use:

```
<ss:link source="$templateSet.searchMembersOnly">
  Members Only Catalog
</ss:link>
```

...which resolves to:

```
<a href=/click/Catalog.bok?mo=1">
  Members Only Catalog
</a>
```

To create a link to a custom template, use:

```
<ss:link source="$templateSet.templates['shortname']">
  Insert Text Here
</ss:link>
```

...which resolves to:

```
<a href=/click/Page.bok?template=shortname">
  Insert Text Here
</a>
```

Attributes

Supports HTML attributes

hidevalue

If the link does not resolve correctly, using this attribute will prevent the link text from displaying on the page.

shortcut

Shortcut attributes allow you to replace a large, complex piece of code with one word. All formatting and text are included within the shortcut. If you want to control formatting or text, replace the shortcut with the corresponding code (for assistance, try viewing the HTML source of the storefront page).

Valid Values

Value	Description
AnchorLink	Creates a link to a reference on the current page.
AnchorName	Creates an <a name> reference.
BuyIt	Creates a link to the Catalog Detail template for the current product, with the buyit.gif image.
BuyItNow	Creates a link that adds the current product to the cart, with the buyitnow.gif image.
GoBack	Creates a link back to the previous template.
ListLongImagesCommon	On the catalog list, displays a “previous” image, “next” image, and a series of identical on/off images for each page, allowing the user to browse the search results. Requires the following images: nextpon.gif, nextpoff.gif, nextnon.gif, nextnoff.gif, nextlon.gif, nextloff.gif. << 1 1 1 1 >> << 1 1 1 1 >> << 1 1 1 1 >>
ListLongImagesUnique	On the catalog list, displays a “previous” image, “next” image, and a series of unique on/off images for each page, allowing the user to browse the search results. Requires the following images: nextpon.gif, nextpoff.gif, nextnon.gif, nextnoff.gif, nextNon.gif, nextNoff.gif (where N is a number for each image in the series, starting with 1). << 1 2 3 4 >> << 1 2 3 4 >> << 1 2 3 4 >>
ListLongText	On the catalog list, displays “Prev”, “Next”, and a series of text numbers for each page. 1 2 3 4 Next Prev 1 2 3 4 Next Prev 1 2 3 4
ListMediumImagesCommon	On the catalog list, displays a series of identical on/off images for each page. Requires the following images: nextlon.gif, nextloff.gif. 1 1 1 1 1 1 1 1 1 1 1 1
ListMediumImagesUnique	On the catalog list, displays a series of identical on/off images for each page. Requires the following images: nextNon.gif, nextNoff.gif (where N is a number for each image in the series, starting with 1). 1 2 3 4 1 2 3 4 1 2 3 4
ListMediumText	On the catalog list, displays a series of text numbers for each page. 1 2 3 4 1 2 3 4 1 2 3 4
ListNext	On the catalog list, displays “Next”. Next Next
ListNext#	On the catalog list, displays a “Next N”, where N represents the number of search results on the next page.

Value	Description
	Next 9 Next 9
ListPrev	On the catalog list, displays “Prev”.
	Prev Prev
ListPrev#	On the catalog list, displays a “Prev N”, where N represents the number of search results on the prior page.
	Prev 9 Prev 9
ListPrevNextImages	On the catalog list, displays “previous” and “next” images. Requires the following images: nextpon.gif, nextpoff.gif, nextnon.gif, nextnoff.gif
	<< >> << >> << >>
ListPrevNextText	On the catalog list, displays “Prev” and “Next”.
	Next Prev Next Prev
ListPrevNextText#	On the catalog list, displays “Prev N” and “Next N”, where N represents the number of search results on the prior and next pages.
	Next 9 Prev 9 Next 9 Prev 9
Lead	Creates a link to display the Lead Product template, with the lead.jpg image.
PoweredBy	Inserts a link to a web site, with the “Powered By” logo.
ShipTrackingNumber	Inserts a link to the Tracking template.

source

The object expression containing the location to which you wish to link.

<ss:media>

Creates an <embed> tag for referencing multimedia files.

Examples

To insert a flash file, use:

```
<ss:media source="intuit.swf" width="550" height="400" type="flash"/>
```

...which resolves to:

```
<object classid='clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B'
  codebase='http://www.apple.com/qtactivex/qtplugin.cab'
  width='550' height='400'>
  <param name='SRC' value='http://url/catalog/Car.avi'>
  <embed src="intuit.swf" width="550" height="400"
    type="application/x-shockwave-flash">
  </embed>
```



```
</object>
```

Attributes

Supports HTML attributes

source (Required)

The object expression that references the media file you wish to embed.

type

The type of file. If this attribute is not present, the tag will attempt to resolve to the correct format using the file extension (for example, files with a *.SWF extension will use the Macromedia Flash player.)

Valid Values

Value	Description
image	Image file (such as *.gif, *.jpg or *.png)
flash	Flash player.
audio	QuickTime player
video	QuickTime player

width

The width of the media file. The use of this attribute is strongly encouraged.

height

The height of the media file. The use of this attribute is strongly encouraged.

<ss:otherwise>

See the <ss:choose> tag for information.

<ss:param>

Passes parameter values to a tag.

Examples

To link to the home page and include the affiliate's referral number, use:

```
<ss:link source="$templateSet.homePage">
  <ss:param name="affiliate_no" value="$affiliate.affiliateNumber"/>
</ss:link>
```

...which resolves to:

```
<a href="../../../StoreFront.bok?affiliate_no=1"></a>
```

Attributes

name (Required)

The name of the parameter

value (Required)

The value of the parameter, can be either a string or a valid object expression.

<ss:password>

Creates an HTML password input field (where the entry is displayed using asterisks for additional security).

Examples

To collect the customer's password, use:

```
<ss:password source="$customer.password" />
```

...which resolves to:

```
<input type="password" name="$customer.password" size="10" maxlength="15">
```

Attributes

Supports HTML attributes

source (Required)

The object expression in which the password will be stored.

value

The default value for the field.

<ss:radio>

Creates a “radio” button field.

Examples

To display a radio button for a shipping method, use:

```
<ss:radio name="shipme" source="$shipQuote.shipMethod" />
```

...which resolves to:

```
<input type="radio" name="shipme" value="S" />
```

Attributes

Supports HTML attributes

checked

Use this to select the radio button by default.

name

The control name. Should only be used if the name is different than the source attribute.

source

The object expression in which the radio button value will be stored.

value

The value to be applied if the radio button is selected.

<ss:select>

Creates a list from which a customer can select an option.

Examples

To create a list of available product colors, use:

```
<ss:select source="$product.colors" name="color" />
```

...which resolves to:

```
<select name="color">
  <option value="Blue">Blue</option>
  <option value="Green">Green</option>
</select>
```

To create a drop-down list of all manufacturers that are currently assigned to products, adding an option to allow the visitor to search for *all* manufacturers, use:

```
<ss:select name="oem" source="$catalog.oemList">
  <option value='all' selected>ALL</option>
</ss:select>
```

...which resolves to:

```
<select name="oem">
  <option value='all' selected>ALL</option>
  <option value="Beauteous Makeup">Beauteous Makeup</option>
  <option value="DollyLand">DollyLand</option>
</select>
```

To create a list of countries that defaults to selecting the user's country, use:

```
<ss:select source="$system.Countries" name="$customer.Country"
  selected="$customer.Country" />
```

...which resolves to:

```
<select name="$customer.country">
```

```
<option value="...">...</option>
<option value="US" selected>United States</option>
<option value="...">...</option>
</select>
```

Attributes

Supports HTML attributes

name (Required)

The object expression in which the selection will be stored.

selected

The value that will be selected by default.

Valid Values

Value	Description
String	Text string
Object Expression	Valid object expression

source (Required)

The object expression containing the data to be displayed in the list.

<ss:set>

Creates an “object” for use on the current template.

Examples

To use the `$product.applyPromotion($promotion)` value repeatedly in code, use:

```
<ss:foreach item="promotion" within="$product.activePromotions">
  <ss:set name="newPrice" value="$product.applyPromotion($promotion)" />
  <ss:if test="$promotion.promotionApplied">
    <ss:if test="$newPrice != $product.finalPrice">
      </ss:if>
    <ss:value source="$newPrice" />
  </ss:if>
</ss:foreach>
```

The `<ss:set/>` tag defines a new object = “`$newPrice`”, and sets its value to “`$product.applyPromotion($promotion)`”. Now, instead of having to type a lengthy object expression repeatedly, you can simply type “`$newPrice`”.

Attributes

name (Required)

The name of the new object.

value (Required)

The object expression that defines the new object.

<ss:sortby>

Determines the sort order of items in a collection specified by the <ss:foreach> tag. You can use multiple <ss:sortby> tags to specify multiple sort orders when used in <ss:foreach within="\$catalog.results">. When used in <ss:foreach within="\$catalog.categoryList()/categoryListWithMembersOnly()/categoriesToDepth()/categoriesToDepthWithMembersOnly()"> (for example, on the Header template), only <ss:sortby item="categorypriority"> may be used (the default for these lists is to sort by category name).

Examples

To sort a list by product name, use:

```
<ss:sortby item="product"/>
```

To sort a list by category priority, then by category, and then by product name, use:

```
<ss:sortby item="categorypriority" direction="asc"/>
<ss:sortby item="category" direction="asc"/>
<ss:sortby item="product" direction="asc"/>
```

Attributes

item (Required)

The item by which a list may be sorted.

Valid Values

Value	Description
category	Category
categorypriority	Category sort priority
oem	Manufacturer Name
price	Product Price
product	Product Name
sku	Product SKU
subcategory	Category

direction

Indicates the direction of the sort order.

Valid Values

Value	Description
asc	Ascending
ascending	Ascending
dsc	Descending

Value	Description
desc	Descending
descending	Descending

<ss:textarea>

Creates a large text box for user entry on a form. If the *rows* or *cols* HTML attributes are not specified, the default will be 4 and 10.

Examples

To collect a message to be sent with an “Email a Friend” referral, use:

```
<ss:textarea source="$email.message" rows="3" cols="40"/>
```

...which resolves to:

```
<textarea name=$email.message cols=40 rows=3 wrap=virtual></textarea>
```

Attributes

Supports HTML attributes

source (Required)

The object expression in which the textarea data will be stored.

<ss:value>

Displays the value of the object. If you are displaying a date object expression, be sure to use the *format* attribute - otherwise, the date will appear as YYYY-MM-DD HH.MM.SS.NNN.

Examples

To display the store name, use:

```
<ss:value source="$store.name"/>
```

...which resolves to:

```
Retroactive
```

To display the product name for every product in the user’s cart, and then limit the length of each name to the first 23 characters, use:

```
<ss:foreach item="detail" within="$cart.details">
  <ss:value source="$detail.product.name" maxlength="23"/><br>
</ss:foreach>
```

...which resolves to:

```
Cutie Doll<br>
Fancy Girl Doll<br>
Dining Room Lithogra...<br>
```

Attributes

source

The object expression containing the value you wish to display.

currency

Defines how currency values should be displayed. The “\$” value should only be used to override Locale formatting.

default

Indicates what should be displayed if no value is found.

escape

If the value being returned by the source attribute contains special characters, use this attribute to indicate how the characters should be escaped. This attribute can be useful when using SSML tags within JavaScript.

Valid Values

Value	Description
javastring	Escapes characters using JavaScript style escaping.
quoted	Escapes characters using quotation marks.
encode	Escapes characters by encoding them.

format

Defines the formatting characteristics for this tag when resolving the “source” attribute.

Valid Values

Value	Description
date.short	Date
date.medium	Date
date.full	Date
date.long	Date
time.short	Time
time.medium	Time
time.full	Time
time.long	Time
time24	Time
day	Day
month	Month
year	Year
percent	Percentage
%	Percentage
%N	Percentage, with the number of decimal places specified by ‘N’
truefalse	True or False

Value	Description
yesno	Yes or No
none	No formatting
money	Currency

maxlength

The maximum number of characters that will be displayed for a value. If this attribute is not specified, the value defined by ProStores will be used.

<ss:when>

Used with <ss:choose> to define a series of test conditions. Test conditions are evaluated in the order in which the <ss:when> tags appear in the code. To provide an alternative if none of the “when” conditions are met, include an <ss:otherwise> tag. See <ss:choose> for more information and examples.

Examples

To apply special formatting to the “shipped” status on an invoice, use:

```
<ss:choose>
  <ss:when test="$detail.status == 'Shipped'">
    <b>This order has been shipped</b>
  </ss:when>
  <ss:otherwise>
    <ss:value source="$detail.status"/>
  </ss:otherwise>
</ss:choose>
```

...which resolves to:

```
<b>This order has been shipped</b>
```

...if shipped. If the order has not been shipped, it displays the current order status (without bold formatting).

Attributes

test (Required)

Defines the test condition. Includes an object expression, operand, and test value. You can test for multiple conditions by using a single “AND” or “OR” statement. AND/OR must either be in all caps, or use the following symbols: “&&” (AND), “||” (OR). Valid operands are:

- ◆ > - Greater Than
- ◆ < - Less Than
- ◆ >= - Greater Than or Equal To
- ◆ <= - Less Than or Equal To
- ◆ = - Equals

- ◆ == - Equals
- ◆ != - Not Equal To
- ◆ <> - Not Equal To
- ◆ MATCHES - Supports wildcard at the end of the string (for example 'string*').

<ss:!doctype>

Appendix D: Tag Mappings

This reference lists all tags used in previous version of ProStores, and their new equivalents. This list includes all tags for all store types and modules.

</loop>

```
</ss:foreach>
```

<::/exclude>

```
</ss:if>
```

<::about>

```
<ss:link source="$templateSet.aboutUs">About Us</ss:link>
```

<::cart>

```
<ss:link source="$templateSet.cart">Shopping Cart</ss:link>
```

<::cartcontents>

1: Standard usage

```
<ss:if test="$cart.details">
  <ss:foreach item="detail" within="$cart.details">
    <ss:value source="$detail.name"/><br>
  </ss:foreach>
<ss:else/>
  Empty
</ss:if>
```

2: When the “maxlength” attribute is used

```
<ss:foreach item="detail" within="$cart.details">
  <ss:value source="$detail.name"
    maxlength="Integer"/><br>
</ss:foreach>
```

<::catalog>

```
<ss:link source="$templateSet.catalog">Catalog</ss:link>
```

<::catalogmbr>

1: Text Only

```
<ss:link source="$templateSet.searchMembersOnly">
  Members Only Catalog
</ss:link>
```

2: Image

```
<ss:link source="$templateSet.searchMembersOnly">
  <ss:image source="$templateSet.images['members.gif']"
    border="0" align="bottom"/>
</ss:link>
```

<::category=CATEGORY>

```
$url.searchlink('category=CATEGORY')
```

<::checkout>

```
<ss:link source="$templateSet.checkout">Checkout</ss:link>
```

<::customtext type="TYPE">

```
<ss:value source="templateSet.customText['TYPE']"/>
```

<::date>	<code><ss:value source="\$system.currentDate" format="date.short" /></code>
<::datelong>	<code><ss:value source="\$system.currentDate" format="date.long" /></code>
<::day>	<code><ss:value source="\$system.currentDate" format="day" /></code>
<::daysto>	<code><ss:value source="\$system.daysTill('DATE') " /></code>
<::daystochristmas>	<code><ss:value source="\$system.daysTill('Christmas') " /></code>
<::download>	<code><ss:link source="\$templateSet.downloadList"> Download Files </ss:link></code>
<::else>	<code><ss:else/></code>
<::emptycart>	<code><ss:link source="\$templateSet.cartEmpty"> Empty Cart </ss:link></code>
<::endif>	<code></ss:if></code>
<::endloop>	<code></ss:foreach></code>
<::exclude>	<code><ss:if test="\$store.storeType != 'STORETYPE'"></code>
<::featuredproduct name="OPTION">	<code><ss:value source="\$store.featuredProduct.OPTION" /></code>
<::fontcolor type="FONTTYPE">	<code><ss:font color="\$font.FONTTYPE"></code>
<::fontface type="FONTTYPE">	<code><ss:font face="\$font.FONTTYPE"></code>
<::fontsize type="FONTTYPE">	<code><ss:font size="\$font.FONTTYPE"></code>
<::goback>	<code><ss:link shortcut="goback" /></code>
<::if <::test=CONDITION>>	<code><ss:if test="CONDITION"></code>
<::if <:hasdownloads>>	<code><ss:if test="\$invoice.hasDownloads"></code>
<::if <:isbrowserversion type="VERSION">>	<code>\$browser.VERSION</code>
<::if <:isfeaturedproduct>>	<code><ss:if test="\$store.featuredProduct"></code>

<::if <:ismember>>
<code><ss:if test="\$customer.isLoggedIn()"></code>
<::if <:ispaymentmethod>>
<code><ss:if test="\$invoice.paymentMethod == PAYMENTMETHOD"></code>
<::if <:ispaymentmethodworldpay>>
<code><ss:if test="\$invoice.paymentMethodCode == WorldPay"></code>
<::if <:issubscription>>
<code><ss:if test="\$product.subscriptionEnabled"></code>
<::if <:issubscriptionrequired>>
<code><ss:if test="\$store.membersOnlySubscriptionRequired"></code>
<::if <:isworldpayfixcontact>>
<code><ss:if test="\$worldpay.isFixContact"></code>
<::if <:isworldpaytestmode>>
<code><ss:if test="\$worldpay.isTestMode"></code>
<::include file=FILE>
<code><ss:include source="\$store.files['FILE']"/></code>
<::include template="TEMPLATE">
<code><ss:include template="TEMPLATE"/></code>
<::join>
1: Image
<code><ss:image source="\$templateSet.images['join.gif']" border="0" align="bottom"/></code>
2: Text
<code><ss:link source="\$templateSet.subscribe">Join Now</ss:link></code>
<::languageoverride>
<code><ss:form shortcut="LanguageOverride"/></code>
<::logoff>
<code><ss:link source="\$templateSet.logoff">Logoff</ss:link></code>
<::logon>
<code><ss:link source="\$templateSet.logon">Logon</ss:link></code>
<::loop=category>
<code><ss:foreach item="category" within="\$catalog.categoriesToDepth(4)"></code>
<::member>
<code><ss:value source="\$customer.fullName"/></code>
<::membercompany>
<code><ss:value source="\$customer.company"/></code>
<::memberfirst>
<code><ss:value source="\$customer.firstName"/></code>
<::memberlast>
<code><ss:value source="\$customer.lastName"/></code>
<::month>
<code><ss:value source="\$system.currentDate" format="month"/></code>
<::oem=MANUFACTURER>
<code>\$url.searchLink('oem=MANUFACTURER')</code>

<::policy>

```
<ss:link source="$templateSet.storePolicies">Policies</ss:link>
```

<::product=PRODUCTNUMBER>

```
<ss:link source="$url.buildFullUrlWithParameter('Detail','no','PRODUCTNUMBER')">
  Text
</ss:link>
```

<::register>

1: Text

```
<ss:image source="$templateSet.images['register.gif']"
  border="0" align="bottom"/>
```

2: Image

```
<ss:link source="$templateSet.customerRegistration">
  Register
</ss:link>
```

<::service>

```
<ss:link source="$templateSet.customerService">
  Customer Service
</ss:link>
```

<::servicehistory>

```
<ss:link source="$templateSet.customerHistory">
  Account History
</ss:link>
```

<::servicelost>

```
<ss:link source="$templateSet.lostPassword">
  Lost Passwords
</ss:link>
```

<::store>

```
<ss:link source="$templateSet.homePage">Home</ss:link>
```

<::storebillboard>

1: Link

```
<ss:link shortcut="$store">
  <ss:image source="$store.getBillboard()" alt="$store.name" border="0"/>
</ss:link>
```

2: Image only

```
<ss:image source="$store.billboard"/>
```

<::storecity>

```
<ss:value source="$store.city"/>
```

<::storecountry>

```
<ss:value source="$store.country"/>
```

<::storedescription>

```
<ss:value source="$store.description"/>
```

<::storeemail>

```
<ss:value source="$store.email"/>
```

<::storefax>

```
<ss:value source="$store.fax"/>
```

<::storeimages>

```
$store.Images
```

<::storelogo>	<code><ss:if test="\$store.isFeatureEnabled('storeLogo')"> <ss:image source="\$storeVersion.storeLogo"/> </ss:if></code>
<::storename>	<code><ss:value source="\$store.name"/></code>
<::storepage=FILE>	<code>\$url.pageFile('FILE')</code>
<::storepage>FILE	<code>\$url.pageFile('FILE')</code>
<::storepath>	<code>\$url.storePath</code>
<::storephone>	<code><ss:value source="\$store.phone"/></code>
<::storescript=SCRIPT>	<code>\$url.buildUrl('SCRIPT')</code>
<::storescriptprefix>	<code>\$system.prefix</code>
<::storescripts>	<code>\$url.basePath</code>
<::storescriptsuffix>	<code>\$system.suffix</code>
<::storestate>	<code><ss:value source="\$store.state"/></code>
<::storestreet>	<code><ss:value source="\$store.street"/></code>
<::storetemplate=TEMPLATE>	<code>\$url.buildUrlWithQueryString('Page', 'template=TEMPLATE')</code>
<::storeurl>	<code><ss:value source="\$store.url"/></code>
<::storezip>	<code><ss:value source="\$store.postalCode"/></code>
<::time24>	<code><ss:value source="\$system.currentDate" format="time24"/></code>
<::time>	<code><ss:value source="\$system.currentDate" format="time.short"/></code>
<::year>	<code><ss:value source="\$system.currentDate" format="year"/></code>
<:addtocart image>	<code><ss:if test="\$product.type == ProductType.LEAD"> <ss:link shortcut="lead"/> <ss:else/> <ss:include template="cartadd"/> </ss:if></code>
<:addtocart template=TEMPLATE>	<code><ss:if test="\$product.type == ProductType.LEAD"></code>

```
<ss:link shortcut="lead"/>
<ss:else/>
  <ss:include template="TEMPLATE"/>
</ss:if>
```

<:addtocart>

```
<ss:if test="$product.type == ProductType.LEAD">
  <ss:link shortcut="lead"/>
<ss:else/>
  <ss:include template="cartadd"/>
</ss:if>
```

<:affiliate>

- 1: Form Field

```
<ss:edit source="$affiliate.name" size="40"/>
```
 - 2: Affiliate Templates

```
<ss:value source="$affiliate.name"/>
```
 - 3: Invoice Templates

```
<ss:value source="$invoice.affiliate.name"/>
```
-

<:affiliatecity>

- 1: Form Field

```
<ss:edit source="$affiliate.city" size="20"/>
```
 - 2: Display Value

```
<ss:value source="$affiliate.city"/>
```
-

<:affiliatecontact>

- 1: Form Field

```
<ss:edit source="$affiliate.contact" size="40"/>
```
 - 2: Display Value

```
<ss:value source="$affiliate.contact"/>
```
-

<:affiliatecountry>

- 1: Form Field

```
<ss:edit source="$affiliate.country" size="40"/>
```
 - 2: Display Value

```
<ss:value source="$affiliate.country"/>
```
-

<:affiliateemail>

- 1: Form Field

```
<ss:edit source="$affiliate.email" size="40"/>
```
 - 2: Display Value

```
<ss:value source="$affiliate.email"/>
```
-

<:affiliatefax>

- 1: Form Field

```
<ss:edit source="$affiliate.fax" size="20"/>
```
 - 2: Display Value

```
<ss:value source="$affiliate.fax"/>
```
-

<:affiliateid>

```
<ss:value source="$invoice.affiliate.affiliateNumber"/>
```

<:affiliateNo>

`<ss:value source="$invoice.affiliate.affiliateNumber" />`

<:affiliatePhone>

1: Form Field

`<ss:edit source="$affiliate.phone" size="20" />`

2: Display Value

`<ss:value source="$affiliate.phone" />`

<:affiliateresale>

1: Display Value

`<ss:value source="$affiliate.resale" />`

2: Form Field

`<ss:edit source="$affiliate.resale" size="20" />`

<:affiliatestate>

1: Form Field

`<ss:edit source="$affiliate.state" size="10" />`

2: Display Value

`<ss:value source="$affiliate.state" />`

<:affiliatestreet1>

1: Form Field

`<ss:edit source="$affiliate.street1" size="30" />`

2: Display Value

`<ss:value source="$affiliate.street1" />`

<:affiliatestreet2>

1: Form Field

`<ss:edit source="$affiliate.street2" size="30" />`

2: Display Value

`<ss:value source="$affiliate.street2" />`

<:affiliatestreet>

1: Form Field

`<ss:edit source="$affiliate.street" size="30" />`

2: Display Value

`<ss:value source="$affiliate.street" />`

<:affiliateurl>

1: Form Field

`<ss:edit source="$affiliate.url" size="40" />`

2: Display Value

`<ss:value source="$affiliate.url" />`

<:affiliatezip>

1: Form Field

`<ss:edit source="$affiliate.postalCode" size="10" />`

2: Display Value

`<ss:value source="$affiliate.postalCode" />`

<:billnew>

```
<ss:checkbox shortcut="ChangePayment" />
```

<:buyit>

```
<ss:link shortcut="BuyIt" />
```

<:buyitnow>

```
<ss:link shortcut="BuyItNow" />
```

<:cancel>

```
<input type="submit" name="cancel" value="Cancel">
```

<:canceledate>

```
<ss:value source="$subscription.cancelDate" format="date.short" />
```

<:cartcheckbox>

```
<ss:if test="$product.attributes">
<ss:else/>
  <ss:checkbox name="cartchk" value="$product.getUrl()" />
</ss:if>
```

<:cartcolor>

```
<ss:if test="$product.hasColors()">
  <ss:select source="$product.colors" name="color" />
</ss:if>
```

<:cartcolortitle>

```
<ss:if test="$product.hasColors()">Color</ss:if>
```

<:cartcustom1>

```
<ss:if test="$product.customService1Title">
  <input type="text" name="custom1" maxlength="50" size="20" />
</ss:if>
```

<:cartcustom1fee>

```
<ss:if test="$product.customService1Title">
  <ss:value source="$product.customService1Fee" />
</ss:if>
```

<:cartcustom1title>

```
<ss:if test="$product.customService2Title">
  <ss:value source="$product.customService1Title" />
</ss:if>
```

<:cartcustom2>

```
<ss:if test="$product.customService2Title">
  <input type="text" name="custom2" maxlength="50" size="20" />
</ss:if>
```

<:cartcustom2fee>

```
<ss:if test="$product.customService2Title">
  <ss:value source="$product.customService2Fee" />
</ss:if>
```

<:cartcustom2title>

```
<ss:if test="$product.customService2Title">
  <ss:value source="$product.customService2Title" />
</ss:if>
```

<:cartend>

```
</ss:form>
```

<:cartname>

1: Form Field

```
<input type="text" name="cartname" size="30" maxlength="30">
```

2: Display Value

```
<ss:value source="$cart.name"/>
```

<:cartqty>

```
<input type="text" name="qty" value="1" size="3">
```

<:cartreset>

```
<input type="reset" value="Clear">
```

<:cartsave image>

```
<ss:if test="$customer.isLoggedIn() AND $cart.details">
  <ss:link source="$templateSet.cartSave">
    <ss:image source="$store.images['cartsave.gif']" border="0" align="bottom"/>
  </ss:link>
</ss:if>
```

<:cartsave>

```
<ss:if test="$customer.isLoggedIn() AND $cart.details">
  <ss:link source="$templateSet.cartSave">
    Save Cart
  </ss:link>
</ss:if>
```

<:cartselect image>

```
<ss:if test="$customer.hasSavedCarts">
  <ss:link source="$templateSet.cartSelect">
    <ss:image source="$store.images['cartsel.gif']"
      border="0" align="bottom"/>
  </ss:link>
</ss:if>
```

<:cartselect>

1: Radio Button

```
<ss:radio name="cartid" value="$cart.cartNumber"/>
```

2: Link

```
<ss:if test="$customer.hasSavedCarts">
  <ss:link source="$templateSet.cartSelect">
    Select Cart
  </ss:link>
</ss:if>
```

<:cartsize>

```
<ss:if test="$product.hasSizes()">
  <ss:select source="$product.sizes" name="size"/>
</ss:if>
```

<:cartsizetitle>

```
<ss:if test="$product.hasSizes()">Size</ss:if>
```

<:cartstart>

```
<ss:form shortcut="CartStart">
```

<:cartsubmit>

```
<input type="submit" value="Add To Shopping Cart">
```

<:category nosub>

```
<ss:select name="category" source="$catalog.categoriesToDepth(0)">
  <option value="all" selected>All</option>
</ss:select>
```

<:category>

```
<ss:value source="$listing.name"/>
```

<:category>

- 1: Storefront template, to display a list of all categories in the store

```
<ss:link source="$category">
  <ss:if test="$category.treeLevel == "1" ">
    <ss:value source="$category.name"/>
  <ss:else/>
    <ss:value source="$category.name"/>
  </ss:if>
</ss:link>
```

- 2: Catalog templates, to display a list of categories

```
<ss:select name="category" source="$catalog.categoryList">
  <option value="all" selected>All</option>
</ss:select>
```

- 3: Within a <ss:foreach item="listing"> tag, displays the category name

```
<ss:value source="$listing.name"/>
```

<:categorydesc>

```
<ss:value source="$category.description"/>
```

<:categoryphoto ALIGN>

```
<ss:image source="$listing.category.photoImage"
  border="0" align="ALIGN"/>
```

<:categoryphoto>

```
<ss:image source="$listing.category.photoImage" border="0"/>
```

<:change name=chgpayment>

```
<ss:form shortcut="ChangeSubscriptionPayment"/>
```

<:change name=chgshipto>

```
<ss:form shortcut="ChangeSubscriptionShipTo"/>
```

<:change>

```
<ss:button source="$templateSet.images['change.gif']" border=0/>
```

<:checkout>

```
<ss:form shortcut="Checkout"/>
```

<:checkoutpro>

```
<ss:form shortcut="CheckoutPro"/>
```

<:city>

- 1: Display Value, customer templates

```
<ss:value source="$customer.city"/>
```

- 2: Form Field, customer templates

```
<ss:edit source="$customer.city" size="20"/>
```

- 3: Form Field, invoice templates

```
<ss:edit source="$address.city" size="20"/>
```

- 4: Display Value, lead templates

```
<ss:value source="$lead.City"/>
```

- 5: Form Field, lead templates

```
<ss:edit source="$lead.City" size="20"/>
```

<:company>

- 1: Display Value, customer templates

```
<ss:value source="$customer.company"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.company" size="30"/>
```

3: Form Field, lead templates

```
<ss:edit source="$lead.company" size="30"/>
```

4: Display Value, lead templates

```
<ss:value source="$lead.company"/>
```

5: Display Value, credit templates

```
<ss:value source="$credit.customer.company"/>
```

<:country>

1: Display Value, customer templates

```
<ss:value source="$customer.country"/>
```

2: Form Field, lead templates

```
<ss:edit source="lead.country" size="30"/>
```

3: Display Value, lead templates

```
<ss:value source="lead.country"/>
```

4: Form Field, customer templates

```
<ss:edit source="$customer.country" size="30"/>
```

5: Form Field, shipping templates

```
<ss:edit source="address.country" size="30"/>
```

<:countrylist>

1: Lead templates

```
<ss:select source="$system.countries" name="$lead.country"
selected="$store.country"/>
```

2: Invoice templates

```
<ss:select source="$system.countries" name="$address.country"
selected="$address.country"/>
```

3: Customer templates

```
<ss:select source="$system.countries" name="$customer.country"
selected="$store.country"/>
```

<:credit>

```
<ss:value source="$customer.storeCredit"/>
```

<:creditdate>

```
<ss:value source="$credit.enterDate" format="date.short"/>
```

<:creditmethod>

```
<ss:value source="$credit.paymentMethod"/>
```

<:creditno>

```
<ss:value source="$credit.creditNumber"/>
```

<:creditnote>

```
<ss:value source="$credit.note"/>
```

<:creditsalestax>

```
<ss:value source="$credit.tax"/>
```

<:creditshipping>

```
<ss:value source="$credit.shipping"/>
```

<:creditsubtotal>

```
<ss:value source="$credit.subtotal"/>
```

<:credittotal>

<ss:value source="\$credit.total"/>

<:currentextension>

<ss:value source="\$extension.currentIndex"/>

<:custcity>

- 1: Display Value, subscription templates

<ss:value source="\$subscription.billToCity"/>

- 2: Display Value, supplier templates

<ss:value source="\$edi.customer.city"/>

- 3: Display Value, customer templates

<ss:value source="\$customer.city"/>

- 4: Display Value, credit templates

<ss:value source="\$credit.customer.city"/>

- 5: Display Value, invoice templates

<ss:value source="\$invoice.billToCity"/>

<:custcompany>

- 1: Display Value, supplier templates

<ss:value source="\$edi.customer.company"/>

- 2: Form Field, customer templates

<ss:edit source="\$customer.company" size="30"/>

- 3: Display Value, customer templates

<ss:value source="\$customer.company"/>

- 4: Display Value, invoice templates

<ss:value source="\$invoice.customer.company"/>

<:custcountry option=full>

- 1: Display Value, customer templates

<ss:value source="\$customer.countryName"/>

- 2: Display Value, subscription templates

<ss:value source="\$subscription.billToCountry"/>

- 3: Display Value, credit templates

<ss:value source="\$credit.customer.country"/>

- 4: Display Value, invoice templates

<ss:value source="\$invoice.shipToCountry"/>

<:custcountry>

- 1: Display Value, subscription templates

<ss:value source="\$subscription.billToCountry"/>

- 2: Display Value, supplier templates

<ss:value source="\$edi.customer.country"/>

- 3: Display Value, customer templates

<ss:value source="\$customer.country"/>

<:custcredit>

<ss:value source="\$customer.storeCredit"/>

<:custcreditcard>

<ss:value source="\$customer.creditCardCode"/>

<:custcreditcardexp>

`<ss:value source="$customer.creditCardExpiration"/>`

<:custcreditcardnumber>

`<ss:value source="$customer.creditCardNumber"/>`

<:custcustom1>

- 1: Display Value, customer templates

`<ss:value source="$customer.customText1"/>`

- 2: Form Field, customer templates

`<ss:edit source="$customer.customText1" size="30"/>`

- 3: Display Value, supplier templates

`<ss:value source="$edi.customer.customText1"/>`

<:custcustom2>

- 1: Display Value, customer templates

`<ss:value source="$customer.customText2"/>`

- 2: Form Field, customer templates

`<ss:edit source="$customer.customText2" size="30"/>`

- 3: Display Value, supplier templates

`<ss:value source="$edi.customer.customText2"/>`

<:custcustom3>

- 1: Display Value, customer templates

`<ss:value source="$customer.customText3"/>`

- 2: Form Field, customer templates

`<ss:edit source="$customer.customText3" size="30"/>`

- 3: Display Value, supplier templates

`<ss:value source="$edi.customer.customText3"/>`

<:custcustom4>

- 1: Display Value, customer templates

`<ss:value source="$customer.customText4"/>`

- 2: Form Field, customer templates

`<ss:edit source="$customer.customText4" size="30"/>`

- 3: Display Value, supplier templates

`<ss:value source="$edi.customer.customText4"/>`

<:custcustom5>

- 1: Display Value, customer templates

`<ss:value source="$customer.customText5"/>`

- 2: Form Field, customer templates

`<ss:edit source="$customer.customText5" size="30"/>`

- 3: Display Value, supplier templates

`<ss:value source="$edi.customer.customText5"/>`

<:custcustom6>

- 1: Display Value, customer templates

`<ss:value source="$customer.custom6"/>`

2: Form Field, customer templates

```
<ss:edit source="$customer.customText6" size="30"/>
```

3: Display Value, supplier templates

```
<ss:value source="$edi.customer.customText6"/>
```

<:custcustom7>

1: Display Value, customer templates

```
<ss:value source="$customer.custom7"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.customText7" size="30"/>
```

3: Display Value, supplier templates

```
<ss:value source="$edi.customer.customText7"/>
```

<:custcustom8>

1: Display Value, customer templates

```
<ss:value source="$customer.custom8"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.customText8" size="30"/>
```

3: Display Value, supplier templates

```
<ss:value source="$edi.customer.customText8"/>
```

<:custemail2>

```
<ss:value source="$customer.email2"/>
```

<:custemail>

1: Display Value, customer templates

```
<ss:value source="$customer.email"/>
```

2: Form field, e-mail a friend templates

```
<ss:edit source="$email.senderEmail"/>
```

<:custfax2>

```
<ss:value source="$customer.workFax"/>
```

<:custfax>

```
<ss:value source="$customer.homeFax"/>
```

<:custfirstname>

```
<ss:value source="$customer.firstName"/>
```

<:custlastname>

```
<ss:value source="$customer.lastName"/>
```

<:custname>

1: Form Field, e-mail a friend templates

```
<ss:edit source="$email.senderName"/>
```

2: Display Value, customer templates

```
<ss:value source="$credit.customer.fullName"/>
```

3: Link, history templates

```
<ss:link source="$customer">  
  <ss:value source="$customer.fullName"/>  
</ss:link>
```

4: Display Value, customer templates

```
<ss:value source="$customer.fullName"/>
```

5: Display Value, subscription templates

```
<ss:value source="$subscription.customer.fullname"/>
```

6: Display Value, supplier templates

```
<ss:value source="$edi.customer.fullName"/>
```

7: Display value, credit templates

```
<ss:value source="$credit.customer.fullName"/>
```

8: Display value, invoice templates

```
<ss:value source="$invoice.customer.fullName"/>
```

<:custno>

1: Display Value, customer templates

```
<ss:value source="$customer.customerNumberFull"/>
```

2: Display Value, supplier templates

```
<ss:value source="$edi.customer.customerNumberFull"/>
```

<:custnotify>

```
<ss:value source="$customer.notify" format="yesno"/>
```

<:custom1>

```
<ss:edit name="custom1" source="$catalog.custom1" size="30"/>
```

<:custom2>

```
<ss:edit name="custom2" source="$catalog.custom2" size="30"/>
```

<:custom3>

```
<ss:edit name="custom3" source="$catalog.custom3" size="30"/>
```

<:custom4>

```
<ss:edit name="custom4" source="$catalog.custom4" size="30"/>
```

<:custpager>

```
<ss:value source="$customer.pager"/>
```

<:custphone2>

1: Display Value, customer templates

```
<ss:value source="$customer.workPhone"/>
```

2: Display Value, invoice templates

```
<ss:value source="$invoice.customer.workPhone"/>
```

<:custphone3>

1: Display Value, customer templates

```
<ss:value source="$customer.mobilePhone"/>
```

2: Display Value, invoice templates

```
<ss:value source="$invoice.customer.mobilePhone"/>
```

<:custphone>

1: Display Value, customer templates

```
<ss:value source="$customer.homePhone"/>
```

2: Display Value, invoice templates

```
<ss:value source="$invoice.customer.homePhone"/>
```

<:custresale>

```
<ss:value source="$customer.resale"/>
```

<:custshipcity>
<code><ss:value source="\$invoice.shipToCity" /></code>
<:custshipcountry>
<code><ss:value source="\$invoice.shipToCountry" /></code>
<:custshipname>
<code><ss:value source="\$invoice.recipient" /></code>
<:custshipstate>
<code><ss:value source="\$invoice.shipToState" /></code>
<:custshipstreet>
<code><ss:value source="\$invoice.shipToStreet" /></code>
<:custshipzip>
<code><ss:value source="\$invoice.shipToPostalCode" /></code>
<:custsince>
<code><ss:value source="\$customer.customerSince" format="yyyy" /></code>
<:custstate>
1: Display Value, supplier templates <code><ss:value source="\$edi.customer.state" /></code>
2: Display Value, customer templates <code><ss:value source="\$customer.state" /></code>
3: Display Value, subscription templates <code><ss:value source="\$subscription.billToState" /></code>
4: Display value, credit templates <code><ss:value source="\$credit.customer.state" /></code>
5: Display value, invoice templates <code><ss:value source="\$invoice.billToState" /></code>
<:custstreet1>
1: Display Value, customer templates <code><ss:value source="\$customer.street" /></code>
2: Display Value, credit templates <code><ss:value source="\$credit.customer.street1" /></code>
<:custstreet2>
1: Display Value, supplier templates <code><ss:value source="\$edi.customer.street2" /></code>
2: Display Value, customer templates <code><ss:value source="\$customer.street2" /></code>
3: Display value, credit templates <code><ss:value source="\$credit.customer.street2" /></code>
4: Display value, subscription templates <code><ss:value source="\$subscription.billToStreet2" /></code>
<:custstreet>
1: Subscription templates <code><ss:value source="\$subscription.billToStreet" /></code> <code><ss:if test="\$subscription.billToStreet2"></code> <code>
<ss:value source="\$subscription.street2" /></code>

```
</ss:if>
```

2: Supplier templates

```
<ss:value source="$edi.customer.street"/>
<ss:if test="$edi.customer.street">
  <br><ss:value source="$edi.customer.street"/>
</ss:if>
```

3: Customer templates

```
<ss:value source="$customer.street"/>
<ss:if test="$customer.street2">
  <br><ss:value source="$customer.street2"/>
</ss:if>
```

4: Credit templates

```
<ss:value source="$credit.customer.street"/>
<ss:if test="$credit.customer.street2">
  <br><ss:value source="$credit.customer.street2"/>
</ss:if>
```

5: Invoice templates

```
<ss:value source="$invoice.billToStreet"/>
<ss:if test="$invoice.billToStreet2">
  <br><ss:value source="$invoice.billToStreet2"/>
</ss:if>
```

<:custtext1>

```
<ss:edit source="$extension.extensionText1" size="40"/>
```

<:custtext2>

```
<ss:edit source="$extension.extensionText2" size="40"/>
```

<:custtext3>

```
<ss:edit source="$extension.extensionText3" size="40"/>
```

<:custtext4>

```
<ss:edit source="$extension.extensionText4" size="40"/>
```

<:custtext5>

```
<ss:edit source="$extension.extensionText5" size="40"/>
```

<:custtext6>

```
<ss:edit source="$extension.extensionText6" size="40"/>
```

<:custtext7>

```
<ss:edit source="$extension.extensionText7" size="40"/>
```

<:custtext8>

```
<ss:edit source="$extension.extensionText8" size="40"/>
```

<:custtext9>

```
<ss:edit source="$extension.extensionText9" size="40"/>
```

<:custzip>

1: Display Value, customer templates

```
<ss:value source="$customer.postalCode"/>
```

2: Display Value, supplier templates

```
<ss:value source="$edi.customer.postalCode"/>
```

3: Display Value, subscription templates

```
<ss:value source="$subscription.billToPostalCode"/>
```

4: Display value, credit templates

```
<ss:value source="$credit.customer.postalCode"/>
```

5: Display value, invoice templates

```
<ss:value source="$invoice.billToPostalCode"/>
```

<:delete>

```
<input type="submit" name="delete" value="Delete"/>
```

<:edidate>

```
<ss:value source="$ediSet.date"/>
```

<:edigscontrolno>

```
<ss:value source="$ediSet.ediGsControlNo"/>
```

<:edigscount>

```
<ss:value source="$ediSet.ediGsCount"/>
```

<:ediisacontrolno>

```
<ss:value source="$ediSet.ediIsaControlNo"/>
```

<:edipolcount>

```
<ss:value source="$edi.ediPolCount"/>
```

<:edipollineno>

```
<ss:value source="$detail.ediPolLineNo"/>
```

<:edistcontrolno>

```
<ss:value source="$edi.ediStControlNo"/>
```

<:edistcount>

```
<ss:value source="$edi.ediStCount"/>
```

<:editime>

```
<ss:value source="$ediSet.time"/>
```

<:email2>

```
<ss:edit source="$customer.email2" size="40"/>
```

<:email>

1: Display Value, customer templates

```
<ss:value source="$customer.email"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.email" size="40"/>
```

3: Form Field, lead templates

```
<ss:edit source="$lead.email" size="40"/>
```

4: Display Value, lead templates

```
<ss:value source="$lead.email"/>
```

5: Form Field, lost password and login templates

```
<ss:edit source="$customerRegistry.email"
value="$customer.email" size="40"/>
```

<:emailafriend>

```
<ss:link source="$templateSet.emailAFriend">
  Email A Friend
</ss:link>
```

<:emailcc>

```
<ss:edit source="$customer.email2" size="40"/>
```

<:emptycart>

```
<ss:if test="$cart.details">
  <ss:link source="$templateSet.cartEmpty">EmptyCart</ss:link>
</ss:if>
```

<:endloop>

```
</ss:foreach>
```

<:errmsg>

```
<ss:value source="$message"/>
```

<:expirationdate>

```
<ss:value source="$subscription.expirationDate"
  format="date.short" default="&nbsp;"/>
```

<:fax2>

1: Display Value, customer templates

```
<ss:value source="$customer.workFax"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.workFax" size="20"/>
```

<:fax>

1: Display Value, customer templates

```
<ss:value source="$customer.homeFax"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.homeFax" size="20"/>
```

3: Display Value, lead templates

```
<ss:edit source="$lead.fax"/>
```

4: Form Field, lead templates

```
<ss:edit source="$lead.fax" size="20"/>
```

<:filelink>

```
<ss:link source="$detail">
  <ss:value source="$detail.downloadFileName">
</ss:link>
```

<:firstname>

1: Display Value, lead templates

```
<ss:value source="$lead.firstName"/>
```

2: Form Field, lead templates

```
<ss:edit source="$lead.firstName" size="20"/>
```

3: Display Value, customer templates

```
<ss:value source="$customer.firstName"/>
```

4: Form Field, lost password and login templates

```
<ss:edit source="$customerRegistry.firstName" size="20"/>
```

5: Form Field, customer templates

```
<ss:edit source="$customer.firstName" size="20"/>
```

<:footer>

```
<ss:include template="footer"/>
```

<:giftwrap>

```
<ss:if test="$store.gift && $store.wrap">
  <ss:checkbox source="$address.giftWrapEnabled"/>
  For an additional <ss:value source="$store.wrapPrice"/>
  we can gift wrap your package.
</ss:if>
```

<:groupbycategory>

```
<ss:sortby item="category"/>
```

<:groupbyoem>	<code><ss:sortby item="oem"/></code>
<:groupbyproduct>	<code><ss:sortby item="product"/></code>
<:groupbysubcategory>	<code><ss:sortby item="category"/></code>
<:header>	<code><ss:include template="header"/></code>
<:if <:isaffiliatepresent>>	<code><ss:if test="\$invoice.affiliate"></code>
<:if <:ischeckout>>	<code><ss:if test="\$invoice.invoiceNumber == 0"></code>
<:if <:ispromodiscountpresent>>	<code><ss:if test="\$invoice.promotionDiscount"></code>
<:if <:isshippingamount>>	<code><ss:if test="\$invoice.hasShippingQuotes()"></code>
<:if <:issubscription>>	<div>1: Test to see if subscriptions are enabled for the store</div> <div><code><ss:if test="\$product.subscriptionEnabled"></code></div> <div>2: Tests to see if the customer has active subscriptions</div> <div><code><ss:if test="\$customer.hasActiveSubscriptions"></code></div>
<:invoiceno>	<div>1: Credit templates</div> <div><code><ss:value source="\$credit.invoiceNumber" default="N/A"/></code></div> <div>2: Invoice templates</div> <div><code><ss:value source="\$invoice.invoiceNumber"/></code></div> <div><code><ss:if test="\$supplier.suffixInd"></code></div> <div><code><ss:value source="\$supplier.suffix"/></code></div> <div><code></ss:if></code></div> <div>3: Supplier templates</div> <div><code><ss:value source="\$edi.invoiceNumber"/></code></div>
<:keyword>	<div>1: Form Field</div> <div><code><ss:edit source="\$catalog.keyword" size="30"/></code></div> <div>2: Display Value</div> <div><code><ss:value source="\$catalog.keyword"/></code></div>
<:keywordoption>	<div><code><input type="radio" name="keywordoption" value="AND" checked>And</code></div> <div><code><input type="radio" name="keywordoption" value="OR">Or</code></div>
<:lastbilldate>	<code><ss:value source="\$subscription.lastInvoiceDate" format="date.short"/></code>
<:lastinvoicedate>	<code><ss:value source="\$subscription.lastInvoiceDate" format="date.short"/></code>

<:lastname>

- 1: Display Value, customer templates
`<ss:value source="$customer.lastName"/>`
 - 2: Form Field, customer templates
`<ss:edit source="$customer.lastName" size="20"/>`
 - 3: Display Value, lead templates
`<ss:value source="$lead.lastName"/>`
 - 4: Form Field, lead templates
`<ss:edit source="$lead.lastName" size="20"/>`
 - 5: Form Field, lost password and login templates
`<ss:edit source="$customerRegistry.lastName" size="20"/>`
-

<:lastrenewdate>

`<ss:value source="$subscription.lastRenewDate" format="date.short"/>`

<:lead>

`<ss:link source="$templateSet.lead">
 <ss:image source="$templateSet.images['lead.gif']"
 border="0" align="bottom"/>
</ss:link>`

<:leadcheck1>

- 1: Display Value
`<ss:value source="$lead.CustomCheckbox1" format="yesno"/>`
 - 2: Form Field
`<ss:checkbox source="$lead.customCheckbox1"/>`
-

<:leadcheck2>

- 1: Display Value
`<ss:value source="$lead.CustomCheckbox2" format="yesno"/>`
 - 2: Form Field
`<ss:checkbox source="$lead.customCheckbox2"/>`
-

<:leadcheck3>

- 1: Display Value
`<ss:value source="$lead.CustomCheckbox3" format="yesno"/>`
 - 2: Form Field
`<ss:checkbox source="$lead.customCheckbox3"/>`
-

<:leadcheck4>

- 1: Display Value
`<ss:value source="$lead.CustomCheckbox4" format="yesno"/>`
 - 2: Form Field
`<ss:checkbox source="$lead.customCheckbox4"/>`
-

<:leadcheck5>

- 1: Display Value
`<ss:value source="$lead.CustomCheckbox5" format="yesno"/>`
 - 2: Form Field
`<ss:checkbox source="$lead.customCheckbox5"/>`
-

<:leadcheck6>**1: Display Value**

```
<ss:value source="$lead.CustomCheckbox6" format="yesno"/>
```

2: Form Field

```
<ss:checkbox source="$lead.customCheckbox6"/>
```

<:leadcustom1>**1: Display Value**

```
<ss:value source="$lead.CustomText1"/>
```

2: Form Field

```
<ss:edit source="$lead.customText1"/>
```

<:leadcustom2>**1: Display Value**

```
<ss:value source="$lead.CustomText2"/>
```

2: Form Field

```
<ss:edit source="$lead.customText2"/>
```

<:leadcustom3>**1: Display Value**

```
<ss:value source="$lead.CustomText3"/>
```

2: Form Field

```
<ss:edit source="$lead.customText3"/>
```

<:leadcustom4>**1: Display Value**

```
<ss:value source="$lead.CustomText4"/>
```

2: Form Field

```
<ss:edit source="$lead.customText4"/>
```

<:leadcustom5>**1: Display Value**

```
<ss:value source="$lead.CustomText5"/>
```

2: Form Field

```
<ss:edit source="$lead.customText5"/>
```

<:leadcustom6>**1: Display Value**

```
<ss:value source="$lead.CustomText6"/>
```

2: Form Field

```
<ss:textarea source="$lead.customText6" rows="4" cols="35"/>
```

<:loop=edi>

```
<ss:foreach item="edi" within="$ediset.ediRecords">
```

<:loop=product>

```
<ss:foreach item="detail" within="$invoice.details">
```

<:loop=subcategory>

```
<ss:foreach item="category"
  within="$system.emptyCategoryList">
```

<:loop=subscription>

```
<ss:foreach item="subscription" within="$customer.subscriptions">
```

<:loop>

- 1: Creates a list of customer invoices

```
<ss:foreach item="detail" within="$customer.accountHistory">
```

- 2: Creates a list of downloadable files

```
<ss:foreach item="invoice" within="$customer.downloadFiles">
```

- 3: Creates a list of search results

```
<ss:foreach item="listing" within="$catalog.results">
```

- 4: Creates a list of details for each item on an invoice

```
<ss:foreach item="detail" within="$invoice.details">
```

- 5: Creates a list of stores purchased by a customer

```
<ss:foreach item="store" within="$store.catalog.stores">
```

- 6: Creates a list of details for each item in a cart

```
<ss:foreach item="detail" within="$cart.details">
```

- 7: Creates a list of store classifications

```
<ss:foreach item="classification"
  within="$store.classifications">
```

- 8: Creates a list of saved carts

```
<ss:foreach item="cart" within="$customer.savedCarts">
```

- 9: Creates a list of relateds

```
<ss:foreach item="product" within="$product.relatedProducts">
```

- 10: Creates a list of items for which the shopper does not have authorization to purchase

```
<ss:foreach item="product" within="$cart.unauthorizedProducts">
```

- 11: Creates a list of details for each item on a supplier template

```
<ss:foreach item="detail" within="$edi.details">
```

- 12: Creates a list of items that are low on inventory

```
<ss:foreach item="product" within="$warehouse.lowInventoryProducts">
```

- 13: Creates a list of credits

```
<ss:foreach item="detail" within="$credit.details">
```

<:mallclassificationdescription>

```
<ss:value source="$classification.description" default="&nbsp;" />
```

<:mallclassificationlink>

```
<ss:link source="$classification">
  <ss:value source="$classification.name" />
</ss:link>
```

<:mallclassificationselect>

```
<ss:select name="classificationid" source="$store.classifications">
  <option value="" selected></option>
</ss:select>
```

<:message>

- 1: Form Field

```
<ss:edit source="$address.message" size="40" />
```

- 2: Display Value

```
<ss:value source="$message" />
```

<:messagearea>**1: E-mail a friend templates**

```
<ss:textarea source="$email.message" rows="3" cols="40"/>
```

2: Shipping templates

```
<ss:textarea source="$address.message" cols="40" rows="3" wrap="virtual"/>
```

<:name>

```
<ss:link source="$templateSet.customerProfile">
  <ss:value source="$customer.firstName"/>
  <ss:value source="$customer.lastName"/>
</ss:link>
```

<:next #>

```
<ss:link shortcut="ListNext#"/>
```

<:next image#>

```
<ss:link shortcut="LongImagesUnique"/>
```

<:next image>

```
<ss:link shortcut="ListPrevNextImages"/>
```

<:next pages>

```
<ss:link shortcut="ListNext#"/>
```

<:next>

```
<ss:link shortcut="ListNext"/>
```

<:nextbilldate>

```
<ss:value source="$subscription.nextBillDate" format="date.short"/>
```

<:noshippingamount>

```
<ss:value source="$invoice.shipping"/>
```

<:notify>

```
<ss:value source="$customer.notify" format="yesno"/>
<ss:checkbox source="$customer.notify"/>
```

<:oem>**1: Display Value, catalog list templates**

```
<ss:if test="$store.name AND $listing.oem">
  <ss:value source="$listing.oem.name"/>
<ss:else/>
  <ss:value source="$product.oem.name"/>
</ss:if>
```

2: Form Field, catalog templates

```
<ss:edit source="$catalog.oem" size="30"/>
```

3: Display Value, catalog detail templates

```
<ss:value source="$product.oem"/>
```

<:oemlist>

```
<ss:select name="oem" source="$catalog.oemlist">
  <option value="all" selected>All</option>
</ss:select>
```

<:oemlogo link>

```
<a href="http://$product.oem.url">
  <ss:image source="$product.oem.logo" border="0"/>
</a>
```

<:oemlogo>

```
<ss:image source="$product.oem.logo"/>
```

<:oemurl link>

```
<a href="http://$product.oem.url">
  <ss:value source="$product.oem.url" />
</a>
```

<:oemurl>

```
<ss:value source="$product.oem.url" />
```

<:orderamount>**1: Apply special formatting if this is a credit amount**

```
<ss:if test="$detail.credit">
  <font color="red">
    <ss:value source="$detail.transactionAmount" />
  </font>
<ss:else/>
  <ss:value source="$detail.transactionAmount" />
<ss:if>
```

2: No special formatting

```
<ss:value source="$detail.transactionAmount" />
```

<:ordercredit>

```
<ss:value source="$invoice.credit" />
```

<:orderdate>**1: Link, invoice templates**

```
<ss:link source="$detail">
  <ss:value source="$detail.date" format="date.short" />
</ss:link>
```

2: Value, invoice templates

```
<ss:value source="$invoice.enterDate" format="date.short" />
```

3: Value, supplier templates

```
<ss:value source="$edi.enterDate" format="yyyyMMdd" />
```

<:ordergift>

```
<ss:if test="$invoice.message">
  Message:
  <ss:value source="$invoice.message" />
</ss:if>
```

<:ordermethod>**1: Payment templates**

```
<ss:value source="$paymentMethod.PaymentMethod"
  default="&nbsp;" />
```

2: Customer templates

```
<ss:value source="$customer.PaymentMethod" />
```

3: Invoice templates

```
<ss:value source="$invoice.paymentMethod" />
```

<:orderno>

```
<ss:value source="$detail.itemNumber" default="&nbsp;" />
```

<:ordernote>

```
<ss:value source="$invoice.orderNote" />
```

<:orderpono>

```
<ss:value source="$invoice.purchaseOrderNumber" />
```

<:orderpromo>

<ss:value source="\$invoice.promotionCode"/>

<:orderpromodiscount>

<ss:value source="\$invoice.promotionDiscount"/>

<:orderpromotext>

<ss:foreach item="promotion" within="\$invoice.promotions">
 <ss:value source="\$promotion.invoiceText"/>

</ss:if>

<:ordersalestax>

<ss:value source="\$invoice.tax"/>

<:ordershipdate>

<ss:value source="\$invoice.shipDate" default=" "
 format="date.short"/>

<:ordershipmethod>

<ss:value source="\$invoice.shipMethodDescription"
 default=" " />

<:ordershipping>

<ss:value source="\$invoice.shipping"/>

<:ordershiptrack>

<ss:value source="\$invoice.shipTrackNumber.number"/>

<:orderstatus>

1: History and tracking templates

<ss:value source="\$detail.status" default=" " />

2: Invoice templates

<ss:value source="\$invoice.status"/>

<:ordersubtotal>

<ss:value source="\$invoice.subtotal"/>

<:ordertotal>

1: Subscription templates

<ss:value source="\$subscription.price"/>

2: Invoice templates

<ss:value source="\$invoice.total"/>

<:ordertotaldue>

1: Invoice templates

<ss:value source="\$invoice.totalDue"/>

2: Subscription templates

<ss:value source="\$subscription.price"/>

<:ordertrackingno>

<ss:value source="\$invoice.tracking" default=" " />

<:ordertype>

<ss:value source="\$detail.activity"/>

<:page=FILENAME.EXT+LINK TEXT>

<ss:link source="\$store.files['FILENAME.EXT']">LINK TEXT</ss:link>

<:pager>**1: Display Value, customer templates**

```
<ss:value source="$customer.pager" />
```

2: Form Field, customer templates

```
<ss:edit source="$customer.pager" size="20" />
```

<:password2>

```
<input type="password" name="confirm" size="10">
```

<:password>**1: Display Value**

```
<ss:value source="$customer.password" />
```

2: Form Field

```
<ss:password source="$customerRegistry.password" size="10" />
```

<:paymentmethod>

```
<ss:value source="$subscription.paymentMethod" />
```

<:paymentoption>

```
<ss:foreach item="paymentMethod"
  within="$store.activePaymentMethods
  ($cart.hasSubscriptionProducts())">
  <tr>
  <td colspan="2">
    <ss:radio name="PaymentMethodCode"
      value="$paymentMethod.paymentMethodCode"
      checked="$paymentMethod.isPaymentDefault()" />
    <ss:value source="$paymentMethod.name" /></td>
  </tr>
  <tr>
  <td width="20"></td>
  <td>
    <table border="0" cellpadding="2" cellspacing="2">
    <ss:foreach item="paymentParameter"
      within="$paymentMethod.paymentParameters($browser)">
    <ss:if test="$paymentMethod.name == 'Credit Card'">
      <ss:if test="$paymentParameter.name == 'CreditCardExpMM'">
        <tr align="left" valign="top">
        <td><ss:value source="$paymentParameter.displayText" /></td>
        <td><ss:value source="$paymentParameter.displayFormField()" />
        </td>
        </tr>
        <ss:else/>
        <ss:if test="$paymentParameter.name == 'CreditCardExpYY'">
        <ss:value source="$paymentParameter.displayFormField()" />
        </td>
        </tr>
        <ss:else/>
        <tr align="left" valign="top">
        <td><ss:value source="$paymentParameter.displayText" /></td>
        <td><ss:value source="$paymentParameter.
          displayFormField()" /></td>
        </tr>
        </ss:if>
        </ss:if>
        </ss:else/>
        <tr align="left" valign="top">
        <td><ss:value source="$paymentParameter.displayText" /></td>
        <td><ss:value source="$paymentParameter.
          displayFormField()" /></td>
        </tr>
        </ss:if>
        </ss:foreach>
      </table>
    </td>
  </tr>
</tr>
```

</ss:foreach>

<:phone2>

- 1: Display Value, customer templates

<ss:value source="\$customer.workPhone" />

- 2: Form Field, customer templates

<ss:edit source="\$customer.workPhone" size="20" />

<:phone3>

- 1: Display Value, customer templates

<ss:value source="\$customer.mobilePhone" />

- 2: Form Field, customer templates

<ss:edit source="\$customer.mobilePhone" size="20" />

<:phone>

- 1: Display Value, customer templates

<ss:value source="\$customer.homePhone" />

- 2: Form Field, customer templates

<ss:edit source="\$customer.homePhone" size="20" />

- 3: Display Value, lead templates

<ss:value source="\$lead.phone" />

- 4: Form Field, lead templates

<ss:edit source="\$lead.phone" size="20" />

<:polimit>

<ss:edit source="\$customer.orderLimit" />

<:product alink>

<ss:link shortcut="AnchorLink" name="\$product.productNumber">

<ss:value source="\$product.name" />

</ss:link>

<:product aname>

<ss:link shortcut="AnchorName" name="\$product.productNumber" />

<:product nolink>

- 1: Display Value

<ss:value source="\$product.name" />

- 2: Display Value

<ss:value source="\$detail.product.name" />

<:product>

- 1: Display Value, catalog detail templates

<ss:value source="\$product.name" />

- 2: Link, catalog list templates

<ss:link source="\$product">

<ss:value source="\$product.name" />

</ss:link>

- 3: Form Field, catalog templates

<ss:edit source="\$catalog.product" size="30" />

- 4: Display Value, invoice templates

<ss:value source="\$detail.product.name" />

5: Display Value, subscription templates

```
<ss:value source="$subscription.product.name"/>
```

<:productattr>

1: Invoice templates

```
<ss:value source="$detail.attributeText"/>
```

2: Subscription templates

```
<ss:value source="$subscription.attributeText"/>
```

<:productbackorder>

```
<ss:if test="$detail.isbackordered"><b>Backordered</b></ss:if>
```

<:productbrief>

```
<ss:value source="$product.brief"/>
```

<:productcost>

1: Catalog detail templates

```
<ss:value source="$product.cost"/>
```

2: Invoice templates

```
<ss:value source="$detail.product.cost"/>
```

<:productcustom1>

```
<ss:value source="$product.customText1"/>
```

<:productcustom2>

```
<ss:value source="$product.customText2"/>
```

<:productcustom3>

```
<ss:value source="$product.customText3"/>
```

<:productcustom4>

```
<ss:value source="$product.customText4"/>
```

<:productcustom5>

```
<ss:value source="$product.customText5"/>
```

<:productcustom6>

```
<ss:value source="$product.customText6"/>
```

<:productdesc>

```
<ss:value source="$product.description"/>
```

<:productno>

1: Subscription Templates

```
<ss:value source="$subscription.product.productNumber"/>
```

2: Catalog list templates

```
<ss:value source="$detail.product.productNumber"/>
```

3: Catalog detail templates

```
<ss:value source="$product.productNumber"/>
```

<:productoem>

```
<ss:value source="$product.oem"/>
```

<:productshipping>

```
<ss:value source="$product.shippingCost"/>
```

<:productphoto LEFT/RIGHT>

```
<ss:image source="$product.photoImage"
  align="LEFT/RIGHT" border="0"/>
```

<:productphoto>

<ss:image source="\$product.photoImage" border="0"/>

<:productphotocart>

<ss:image source="\$catalog.images[\$detail.product.thumbnailImageName]"
width="25" height="25"/>

<:productprice>

1: Catalog detail templates

<ss:include macro="productprice"/>

2: Catalog list, invoice templates

<ss:include macro="productprice" />

3: Subscription templates

<ss:value source="\$subscription.price"/>

<:productpriceretail>

<ss:value source="\$detail.product.retailPrice"/>

<:productpricesale>

<ss:value source="\$product.finalPrice"/>

<:productpricesavings format=percent fractiondigits=XX>

<ss:value source="\$product.priceSavingsPercent" format="%XX"/>

<:productpricesavings format=percent>

<ss:value source="\$product.priceSavingsPercent" format="percent"/>

<:productpricesavings>

1: Catalog detail templates

<ss:value source="\$product.priceSavingsCurrency"/>

2: Invoice, catalog list templates

<ss:value source="\$detail.product.priceSavingsCurrency"/>

<:productpricex>

<ss:include macro="attributedetailfull"/>

<:productqty>

<ss:value source="\$detail.quantity"/>

<:productquantity>

1: Subscription templates

<ss:value source="\$subscription.quantity"/>

2: Catalog detail, Catalog List templates

<ss:value source="\$product.quantityInStock"/>

3: Form field, cart template

<ss:edit source="\$detail.quantity" name="qty" size="4"/>

<:productretail>

<ss:value source="\$product.retailPrice"/>

<:productsaleend>

<ss:if test="\$product.onSale">
Sale Ends: <ss:value source="\$product.saleEndDate"
format="date.short"/>
</ss:if>

<:productsku>

- 1: Catalog list templates
`<ss:value source="$detail.sku"/>`
 - 2: Invoice templates
`<ss:value source="$detail.productInfo.sku"/>`
 - 3: Catalog detail templates
`<ss:value source="$product.sku"/>`
-

<:productstock>

- 1: Catalog list templates
`<ss:if test="$detail.product.quantityInStock">
 In Stock
<ss:else/>
 Backordered
</ss:if>`
 - 2: Catalog detail templates
`<ss:if test="$product.quantityInStock">
 In Stock
<ss:else/>
 Backordered
</ss:if>`
-

<:productstockattr>

```
<ss:foreach item="attr" within="$product.attributes">  
    <ss:value source="$attr.size"/>  
    (<ss:value source="$attr.quantityInStock"/>),  
</ss:foreach>
```

<:productstockback>

```
<ss:if test="$product.quantityInStock < 0">Backordered</ss:if>
```

<:productstockimage>

```
<ss:if test="$product.quantityInStock">  
    <ss:image source="$templateSet.images['instock.gif']"   
        border="0"/>  
</ss:if>
```

<:productstockqty>

```
<ss:value source="$product.quantityInStock"/>
```

<:productstockyesno>

```
<ss:value source="$product.quantityInStock" format="yesno"/>
```

<:productsurcharge>

```
<ss:value source="$product.surcharge"/>
```

<:productsurchargeTitle>

```
<ss:value source="$product.surchargeTitle"/>
```

<:productthumbnail link>

```
<ss:link source="$product.photoImage">  
    <ss:image source="$catalog.images[$product.thumbnailImageName]"   
        border="0"/>  
</ss:link>
```

<:productthumbnail nolink>

```
<ss:image source="$catalog.images[$product.thumbnailImageName]"   
    border="0"/>
```

<:productthumbnail>

```
<ss:link source="$product">
```

```
<ss:image source="$product.thumbnailImage" border="0"/>
</ss:link>
```

<:producttotal>

```
<ss:value source="$detail.total"/>
```

<:productunit>

1: Catalog detail templates

```
<ss:value source="$product.unitOfMeasure"/>
```

2: Invoice, catalog list templates

```
<ss:value source="$product.unitOfMeasure"/>
```

<:productweight>

1: Invoice, catalog list templates

```
<ss:value source="$detail.product.weightText"/>
```

2: Catalog detail templates

```
<ss:value source="$product.weight"/>
```

<:promo>

```
<input type="text" name="promo" value="" size="10"/>
```

<:purchasedate>

1: Link

```
<ss:link source="$url.buildUrl('CustomerSubscription')">
  <ss:param name="subscription"
    value="$subscription.subscriptionNumber"/>
  <ss:value source="$subscription.purchaseDate"
    format="date.short"/>
</ss:link>
```

2: Display Value

```
<ss:value source="$subscription.purchaseDate" format="date.short"/>
```

<:recipient>

1: Invoice templates

```
<ss:edit source="$address.recipient" size="40"/>
```

2: E-mail a friend templates

```
<ss:edit source="$email.recipientName"/>
```

<:recipientemail>

1: E-mail a friend templates

```
<ss:edit source="$email.recipientEmail"/>
```

2: Invoice templates

```
<ss:edit source="$email.recipientEmail"/>
```

<:referer>

```
<ss:link source="$templateSet.searchResults">
  Return To Catalog
</ss:link>
```

<:related>

```
<ss:if test="$product.hasRelateds">
  <ss:include template="catalogrelated"/>
</ss:if>
```

<:resale>

1: Display Value, customer templates

<ss:value source="\$customer.resale"/>

2: Form Field, customer templates

<ss:edit source="\$customer.resale" size="20"/>

<:resaleind>

<ss:checkbox source="resale"/>

<:reset>

<input type="reset" value="Clear"/>

<:searchend>

<ss:value source="\$catalog.end"/>

<:searchstart>

<ss:value source="\$catalog.start"/>

<:searchstorecriteria>

<ss:value source="\$storecatalog.searchCriteria" default="All"/>

<:searchstorename>

<ss:link shortcut="MallStoreLink">
 <ss:value source="\$store.name"/>
</ss:link>

<:searchtext>

<ss:value source="\$catalog.searchtext" default="All"/>

<:searchtotal>

<ss:value source="\$catalog.total"/>

<:selectshipping>

<ss:include template="selectshipping"/>

<:shipmethod>

<ss:value source="\$invoice.shipMethodDescription"/>

<:shipnextday>

<input type="radio" name="shipme" value="N"/>
<ss:value source="\$store.shipping.getServiceDescription('N')"/>

<:shipnextdayrate>

<ss:value source="\$cart.getShipping('N')"/>

<:shipping>

<ss:value source="\$cart.shipping" default="0"/>

<:shippingquotes>

<ss:foreach item="quote" within="\$invoice.shippingQuotes">
 <ss:if test="\$invoice.shipMethod == \$quote.shipMethod">
 <option value="\$quote.value" selected>
 <ss:else/>
 <option value="\$quote.value">
 <ss:value source="\$quote.name"/>
 <ss:value source="\$quote.amount/>
 </option>

<:shipstandard>

<input type="radio" name="shipme" value="S" checked/>
<ss:value source="\$store.shipping.getServiceDescription('S')"/>

<:shipstandardrate>

<ss:value source="\$cart.getShipping('S')"/>

<:shipto>

<ss:checkbox shortcut="ChangeShippingAddress"/>

<:shiptocity>

1: Display Value

<ss:value source="\$invoice.shipToCity"/>

2: Display Value, subscription templates

<ss:value source="\$subscription.shipToCity"/>

<:shiptocompany>

1: Display Value

<ss:value source="\$invoice.company"/>

2:

<ss:value source="\$invoice.company" default=" "/>

<:shiptocountry option=full>

1: Invoice templates

<ss:value source="\$invoice.shipToCountry.name"/>

2: Subscription templates

<ss:value source="\$subscription.shipToCountry.name"/>

<:shiptocountry>

1: Display Value, subscription templates

<ss:value source="\$subscription.shipToCountry"/>

2: Display Value

<ss:value source="\$invoice.shipToCountry"/>

<:shiptoname>

1: Subscription templates

<ss:value source="\$subscription.recipient"/>

2: Invoice templates

<ss:value source="\$invoice.recipient"/>

<:shiptophone>

1: Subscription templates

<ss:value source="\$subscription.phone"/>

2: Invoice templates

<ss:value source="\$invoice.phone"/>

<:shiptostate>

1: Subscription templates

<ss:value source="\$subscription.shipToState"/>

2: Invoice templates

<ss:value source="\$invoice.shipToState"/>

<:shiptostreet2>

<ss:value source="\$invoice.shipToStreet2"/>

<:shiptostreet>**1: Subscription templates**

```
<ss:value source="$subscription.shipToStreet" />
<ss:if test="$subscription.shipToStreet2">
  <br/><ss:value source="$subscription.shipToStreet2" />
</ss:if>
```

2: Invoice templates

```
<ss:value source="$invoice.shipToStreet" />
<ss:if test="$invoice.street2">
  <br/><ss:value source="$invoice.street2" />
</ss:if>
```

<:shiptozip>**1: Display Value, subscription templates**

```
<ss:value source="$subscription.shipToPostalCode" />
```

2: Display Value

```
<ss:value source="$invoice.shipToPostalCode" />
```

<:shiptwoday>

```
<input type="radio" name="shipme" value="2" />
<ss:value source="$store.shipping.getServiceDescription('2')"/>
```

<:shiptwodayrate>

```
<ss:value source="$cart.getShipping('2')"/>
```

<:sortorder primary="ITEM">

```
<ss:sortby item="ITEM" />
```

<:sortorder secondary="ITEM">

```
<ss:sortby item="ITEM" />
```

<:sortorder tertiary="ITEM">

```
<ss:sortby item="ITEM" />
```

<:state>**1: Form Field, invoice templates**

```
<ss:edit source="$address.state" size="10" />
```

2: Form Field, lead templates

```
<ss:edit source="$lead.state" size="10" />
```

3: Display Value, lead templates

```
<ss:edit source="$lead.state" />
```

4: Display Value, customer templates

```
<ss:value source="$customer.state" />
```

5: Form Field, customer templates

```
<ss:edit source="$customer.state" size="10" />
```

<:statelist country="COUNTRY" default="DEFAULT">**1: Customer, Checkout templates**

```
<ss:select source="$system.states('COUNTRY') "
  name="$customer.state" selected="DEFAULT" />
```

2: Ship To templates

```
<ss:select source="$system.states('COUNTRY') "
  name="$address.state" selected="DEFAULT" />
```

3: Lead templates.

```
<ss:select source="$system.states('COUNTRY') "
```

```
name="$lead.state" selected="DEFAULT"/>
```

<:statelist country="COUNTRY">

1: Customer, Checkout templates

```
<ss:select source="$system.states('COUNTRY') "
name="$customer.state"/>
```

2: Ship To templates

```
<ss:select source="$system.states('COUNTRY') "
name="$address.state"/>
```

3: Lead templates.

```
<ss:select source="$system.states('COUNTRY') "
name="$lead.state"/>
```

<:status>

```
<ss:value source="$subscription.status"/>
```

<:storelist>

```
<ss:select source="$customer.stores()" name="custom1"/>
<option value="Select Store" selected>--Select Store--</option>
</ss:select>
```

<:storesaletext>

```
<ss:foreach item="promotion"
within="$store.activeStorePromotions">
<ss:if test="$store.isValidPromotion($promotion)">
<ss:value source="$promotion.invoiceText"/><br>
</ss:if>
</ss:foreach>
```

<:street1>

1: Display Value, customer templates

```
<ss:value source="$customer.street"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.street" size="30"/>
```

3: Form Field, lead templates

```
<ss:edit source="$lead.street" size="30"/>
```

4: Form Field, invoice templates

```
<ss:edit source="$address.street" size="30"/>
```

<:street2>

1: Display Value, customer templates

```
<ss:value source="$customer.street2"/>
```

2: Form Field, customer templates

```
<ss:edit source="$customer.street2" size="30"/>
```

3: Form Field, invoice templates

```
<ss:edit source="$address.street2" size="30"/>
```

4: Display Value, lead templates

```
<ss:value source="$lead.street2"/>
```

5: Form Field, lead templates

```
<ss:edit source="$lead.street2" size="30"/>
```

<:street>

1: Display Value, lead templates

```
<ss:value source="$lead.Street"/>
```

2: Display Value, customer templates

```
<ss:value source="$customer.street"/>
```

3: Form Field, customer templates

```
<ss:edit source="$customer.street" size="30"/>
```

<:subcategory>

1: Display Value

```
<ss:value source="$listing.name"/>
```

2: Link

```
<ss:link source="$url.buildUrl('Search')">
  <ss:param name="category" value="$category.parentChainDisplay"/>
  <ss:value source="$category.name"/>
</ss:link>
```

<:subject>

```
<ss:emailheader>
  Enter Text Here
</ss:emailheader>
```

<:submit>

1: Subscription templates

```
<ss:form shortcut="SubmitSubscription"/>
```

2: All other templates

```
<input type="submit" name="submit" value="Submit">
```

3: Catalog Detail template

```
<input type="submit" name="submit" value="Add to Shopping Cart">
```

<:subscribe>

```
<input type="submit" value="Subscribe">
```

<:subscriptionbillnote>

```
<ss:foreach item="subscription" within="$invoice.subscription"/>
  <p><ss:value source="$subscription.billNote"/></p>
</ss:foreach>
```

<:subscriptionno>

```
<ss:value source="$subscription.subscriptionNumber"/>
```

<:subscriptionperiod>

1: Subscription templates

```
<ss:value source="$subscription.subscriptionPeriodText" default="&nbsp;"/>
```

2: Catalog detail templates

```
<ss:if test="$product.subscriptionEnabled">
  <ss:select source="$product.subscriptions" name="productattr"/>
</ss:if>
```

<:subtotal>

```
<ss:value source="$cart.subtotal" default="0"/>
```

<:supplier>

```
<ss:value source="$supplier.name"/>
```

<:suppliercity>

```
<ss:value source="$supplier.city"/>
```

<:supplieredicode>

```
<ss:value source="$supplier.ediCode"/>
```

<:supplierstate>	<code><ss:value source="\$supplier.state"/></code>
<:supplierstreet>	<code><ss:value source="\$supplier.street"/></code>
<:supplierzip>	<code><ss:value source="\$supplier.postalCode"/></code>
<:taxcounty>	<code><ss:select source="\$system. taxCounties(\$address.country, \$address.postalCode)" name="county"/></code>
<:taxregionlist>	<code><ss:select name="taxregion" source="\$store.taxRegionsSelect" selected="\$customer.defaultTaxCode"></code>
<:taxregionselect>	<code><ss:include template="TaxRegionSelect"/></code>
<:total>	<code><ss:value source="\$cart.total" default="0"/></code>
<:totalextensions>	<code><ss:value source="\$extension.total"></code>
<:tracking>	<code><ss:form shortcut="Tracking"/></code>
<:trigger>	<code><input type="hidden" name="trigger-post" value="VALUE"></code>
<:worldpay name="OPTION">	<code><ss:value source="\$worldpay.OPTION"/></code>
<:zip>	<div>1: Form Field, lead templates</div> <div><code><ss:edit source="\$lead.postalCode" size="10"/></code></div> <div>2: Display Value, lead templates</div> <div><code><ss:value source="\$lead.postalCode"/></code></div> <div>3: Form Field, invoice templates</div> <div><code><ss:edit source="\$address.postalCode" size="10"/></code></div> <div>4: Display Value, customer templates</div> <div><code><ss:value source="\$customer.postalCode"/></code></div> <div>5: Form Field, customer templates</div> <div><code><ss:edit source="\$customer.postalCode" size="10"/></code></div>

A

About Us Template	87
Advanced Editor.....	37
advanced fonts and colors	67
attributedetailfull	70
attributes	
hide	71
show on sale.....	75
Auction Email Template	87

B

background colors, change in store.....	61
baselines	
revert.....	21
set all templates	20
Basic Editor	37
borders, template	39
browse mode	12
business objects overview	9

C

cart	
show total items	76
cart template	
display related products	75
Cart Templates	
Cart	88
Cart Add	88
Cart Save.....	88
Cart Select.....	88
Cascading Style Sheets	56
Catalog List Templates	
Catalog List.....	90
Catalog List - Members Only	90
Catalog List - Not Found.....	90
Catalog Templates	
Catalog	88
Catalog Detail	89
Catalog Detail - Members Only	89
Catalog, Members Only.....	89
Related Products	89
categories	
custom templates	59
upload images	50
category list	
change display	59

category lists	
collapsible.....	70
comparison	74
Checkout Template	
Deadend.....	91
New Customers	91
Start	90
Closed Template	91
code samples	66
code snippet	
featured products	66
codes, show on templates	39
colors	
change in theme.....	61
change label in storefront.....	59
eBay Store	67
conditions.....	30
context menu.....	39
copy center	12
County Tax Template	92
Create Custom Links to Your Store page	14
Credit Templates	
Email	92
HTML.....	92
CSS	56
custom templates.....	59
Customer History Template.....	92
Customer Properties Template	92
Customer Service Template.....	93
Customer Sign In Template	93
Customer Subscription Template	93

D

data	28
design mode	34
design mode editing	37
Basic Editor	37
design settings	
revert to baseline	21
revert to original version	21
design, importance of.....	4
documents, additional	4
Download List Template	93
drill down	
Catalog List	72
dynamic pages.....	5

E

eBay Email Template	87
eBay Store	
fonts and colors	67
edit	
page templates	18
rules	46
status	49
store version language/local	46
editor toolbar	38
editors	18
Advanced	37
Basic	37
Source Editor	42
elements	
overview	41
Email A Friend Content Template	94
Email A Friend Results Template	94
Email A Friend Template	93
existing sites	
transfer look and feel	10
working with ProStores	10
export	
page template	33

F

featured categories code snippet	66
file management	57
files	
working with	55
files, upload	55
fonts	
eBay Store	67
fonts, change in store	61
footer	
hide	7
hide, examples	58
Footer Template	94
forms	
quick search	69
Frequently Asked Questions Templates	94
FTP area	57

G

Generated Account Password - Email	
Template	95
glossary	83
guidelines, template	39

H

handlebars	22
header	

hide	7
hide, examples	58
Header Template	95

I

images	27
category, upload	50
customizable	54
mail, upload	50
manufacturer, upload	50
product	53
product, upload	50
shared images	50
store	53
store version images	50
submit button	54
troubleshooting	79
upload	50
working with	50
import	
page template	33
introduction	4
Invoice Template	
Email	96
HTML	95
Print	96
invoice template, display order extensions	72

J

JavaScript	57
------------------	----

K

keyboard shortcuts	39
--------------------------	----

L

link to external pages	16
links	25
managing	16
Listing Email Template	87
lists	29
logo	
add to store	61
Lost Password Templates	
E-mail	97
HTML	96
Low Inventory - Email Template	97

M

manager	
links	16
manufacturers	

custom templates	59
upload images	50
markers, show on templates	39
Member Login Templates	
Error Message	97
Login	97
Message Template	98
meta-data	
add to store	61
meta-tags, creating on static page	55
migrate site	12
multimedia files	57
multimedia files, upload	55
multiple templates	60
overview	59

O

objects	
overview	9
OEM	
custom templates	59
upload images	50
Order Confirm Template	98
order extensions	72
Order Extensions Template	98

P

package tracking	106
Packing Slip Template	98
page template	
export	33
import	33
Page Template Editor	18
pages	
publish	21
pages, troubleshooting	80
PayPal Purchase Token Templates	99
Policies Template	99
price formatting	70
Privacy Template	99
Product Authorization Template	99
product images	53
products	
add to cart	69
custom templates	59
duplicate search results	81
upload images	50
Promotion Alert Templates	100
publish	
pages	21
Purchase Order Limit Template	100

Q

quick search form	69
-------------------------	----

R

Recurring Payment Template	100
redesign store	61
Registration Templates	
Confirmation	101
Register	101
related products	
display on cart	75
resale	
add to Checkout New Customer template	91
add to Checkout template	90
revert	
original version	21
revert to baseline	21
revert to original version	21
right-click menu	39
rules	
edit	46

S

Sales Affiliate Templates	
Confirmation	101
Email	101
Registration	101
Sales Lead Templates	
Confirmation	102
Lead	102
select content	22
Select Shipping Template	102
server file management	57
services	
add to cart	69
custom templates	59
upload images	50
shared images	50
Ship To Templates	
Ship To	102
Ship to Gift	103
shipping	
suppress method	72
Shipping - Email Template	103
shortcut menu	39
show borders on templates	39
show codes on templates	39
show markers on templates	39
site migration	12
sizes	
change label in storefront	59
sorting	

legacy tags	40
source code, editing templates	42
Source Editor	42
source mode	34
spiders	55
splash pages, create	55
static files	
create	55
css	56
JavaScript	57
multimedia	57
upload	55
working with	55
static pages, create	55
status	
edit	49
store	
versions	44
store content area	57
store data area	57
Store Design	
adding images	27
adding links	25
adding text	25
building lists	29
conditions	30
display store data	28
store design, importance of	4
store images	53
Store Locator Template	103
store logo, upload	50
store version	
rules	46
store version images	50
store version language	
edit	46
store version locale	
edit	46
store versions	43
understanding	43
Storefront Template	103
stylesheets	56
subcategories	72
submit button image	54
Subscription Templates	
Cancellation Email	104
Recurring Bill Error - Email	100
Renewal Email	104
Start	104
Supplier Templates	
EDI	104
EDI 850	105
E-mail	105
Fax	105

T

tag reference	
conversion	137
mapping	137
overview	8
reference	108
Tax Region Select Template	105
template sets	
now Store Versions	43
templates	
baselines, set for all templates	20
context menu	39
copy from web site	12
design mode	37
edit	18
editing source code	42
elements overview	41
export	33
how built	7
how displayed	5
import	33
keyboard shortcuts	39
manage work	19
multiple	60
right-click menu	39
saving	32
shortcut menu	39
show borders	39
show markers	39
toolbar	38
undoing	32
user defined	61
validate	32
viewing	34
visual aids	39
working with	18
test plans	77
testing	77
browsers	77
catalog templates	78
checkout templates	78
customer service templates	79
equipment	77
membership	78
operating systems	77
plans	77
product templates	78
screen resolution	77
store templates	77
text	25
themes	
change colors	61
choosing	10
customize	11

tracking packages	106
Tracking Template	106
troubleshooting	79
images	79
pages	80
validation	79

U

undo template changes	61
upload	
images	50
multimedia	55
static files	55
store logo	50
UPS, tracking link	69
user defined templates	61
User Defined Templates	106

V

validate	
page templates	32
validation	
troubleshooting	79
versions	
store	44
view, set defaults	34

W

WorldPay Templates	
Cancel	107
Confirm	107
Error	106
Purchase Token	107
WYSIWYG editing	37
Basic Editor	37