

# STS 4.3 with HTC 2.5.9

Updated as of November 4, 2006

## Table of contents

<b>1</b>	<b>BASICS ABOUT STS.....</b>	<b>3</b>
1.1	WHAT IS IT? .....	3
1.2	HOW STS WORKS (BASIC) .....	3
<b>2</b>	<b>INSTALLATION.....</b>	<b>3</b>
<b>3</b>	<b>HOW TO USE STS V4.3:.....</b>	<b>3</b>
3.1	SIMPLE, ONE PAGE TEMPLATE.....	3
3.2	ADD NEW BOXES OR TEMPLATE VARIABLES .....	4
3.2.1	<i>Add new variables in sts_user_code.php</i> .....	4
3.2.2	<i>Add new variables in a new file</i> .....	4
3.3	CHANGE THE NAME OF THE DEFAULT TEMPLATE .....	4
3.4	CHANGE THE FOLDER WHERE TEMPLATES ARE LOCATED .....	5
3.4.1	<i>Change the design of the shop</i> .....	5
3.4.2	<i>Try a new design only for the administrator</i> .....	5
3.5	DEBUG MODE .....	5
3.6	CREATE A TEMPLATE FOR A SPECIFIC PAGE .....	5
3.7	CREATE TEMPLATES FOR SPECIFIC CATEGORIES .....	6
3.8	CREATE TEMPLATES FOR MANUFACTURERS .....	7
3.9	CREATE TEMPLATES FOR SPECIFIC PRODUCTS .....	8
3.9.1	<i>Template for header, footer and columns</i> .....	8
3.9.2	<i>Template for the content of the page</i> .....	9
3.9.3	<i>Why a content template?</i> .....	10
3.9.4	<i>STS 3 compatibility mode</i> .....	10
<b>4</b>	<b>LIST OF EXISTING PLACEHOLDERS .....</b>	<b>11</b>
4.1	GENERAL PLACEHOLDERS .....	11
4.2	PLACEHOLDERS FOR INFOBOXES .....	12
4.3	PLACEHOLDER FOR MAIN CONTENT.....	13
4.4	SPECIAL PLACEHOLDER FOR HTML HEADER .....	13
4.5	PLACEHOLDERS FOR FOOTER.....	15
4.6	PLACEHOLDER FOR USER CREATED TAGS .....	15
4.7	PLACEHOLDERS FOR PRODUCT_INFO.PHP CONTENT TEMPLATE.....	16
4.8	PLACEHOLDERS FOR PRODUCT_INFO.PHP TEMPLATE IN COMPATIBILITY MODE.....	17
<b>5</b>	<b>TIPS AND TRICKS.....</b>	<b>17</b>
5.1	ABSOLUTE URLs .....	17
5.2	SAVE SOME EXECUTION TIME .....	18
5.2.1	<i>Change placeholder names</i> .....	18
5.2.2	<i>Unused code</i> .....	18
5.3	CREATE YOUR OWN BUTTONS (OPTIONAL) .....	19
5.4	CREATE YOUR OWN IMAGES (OPTIONAL) .....	20
5.5	REPLACE PAGE HEADING TITLE WITH THE NAME OF CATEGORIES OR SUBCATEGORIES (OPTIONAL).....	21
5.6	CREATE YOUR OWN STS MODULES (OPTIONAL).....	22
5.6.1	<i>Why create STS modules?</i> .....	22
5.6.2	<i>How?</i> .....	22
5.7	CREATE YOUR OWN PLACEHOLDER FILE TO INCLUDE (OPTIONAL) .....	23
5.7.1	<i>Why?</i> .....	23
5.7.2	<i>How?</i> .....	23

<b>6</b>	<b>INFOBOX TEMPLATES .....</b>	<b>24</b>
6.1	DESCRIPTION.....	24
6.2	TEST.....	26
<b>7</b>	<b>SUPPORT FORUM.....</b>	<b>26</b>

# 1 Basics about STS

## 1.1 What is it?

STS is an easy to install template system for osCommerce 2.2MS2. The graphical part of the shop can be designed using a standard html editor like Dreamweaver or FrontPage.

## 1.2 How STS works (basic)

Instead of displaying the result of the scripts straight to the browser, STS "captures" it into variables. To each variable corresponds a placeholder (for example "\$banner") that can be placed on an html template. STS will then replace all placeholders with the corresponding code and display it in the browser. This involves only a few modifications of the osCommerce files which allow you to change the design of the shop by simply modifying an html page.

In its most simple form, only one template is created and it will be used for all pages of the shop. It is also possible to create a template for a specific page, a specific category, a specific manufacturer, a specific product and now in version 4.3; a specific infobox.

I have included a "blank" template to show what code is required in your HTML pages for you to make your own templates. Using this blank template, you can insert the various variables (placeholders) that you desire to use in your store to match your HTML design. This template is located in "catalog/includes/sts\_templates/test/sts\_blank\_template.html".

# 2 Installation

Refer to separate document for the installation. After installation, STS is not enabled and your shop looks the same as before. Continue to read from here once you have completed the installation.

# 3 How to use STS v4.3:

## 3.1 Simple, One Page Template

From the admin panel in your osCommerce store, go to modules – STS. Install and enable the module named "Default". The look of your shop will now change to the look of the template that is located in:

- /catalog/includes/sts\_templates/test/sts\_template.html

Use style sheet settings or HTML settings to provide a custom look and feel to your site and arrange elements (placeholders) wherever you like them.

- See chapter 3.3 if you want to change the name of the template.
- See chapter 3.4 if you want to change the folder where the templates are located.
- See chapter 3.6 for making templates for specific pages.
- See chapter 4 for a list of available placeholders.

## **3.2 Add new boxes or template variables**

### **3.2.1 Add new variables in sts\_user\_code.php**

If you want to add any new boxes or template variables, add them in the `includes/modules/sts_inc/sts_user_code.php` file. An example of syntax is provided.

- See chapter 5.7 for another way to create new variables.

### **3.2.2 Add new variables in a new file**

It is also possible to create new variables in a new file:

- Create a new php file containing the code for new variables. An example of code is available in file:  
“`includes/modules/sts_inc/sts_user_code.php`”
- Put this new file in the following folder:  
“`includes/modules/sts_inc/`”
- Go to admin panel and edit module "Default": add this new file name after `sts_user_code.php`.
- There can be several names defined here. They must be separated by a semicolon (example: `sts_user_code.php;header_tags.php`).

## **3.3 Change the name of the default template**

As you will see later, it is possible to have templates for some specific pages. The default template is the one used if there is no specific template found for the page being displayed.

After a fresh install, the default template is located here:

- `/catalog/includes/sts_templates/test/sts_template.html`

The name of the default template in use can be changed by editing the "Default" module from the admin panel.

### ***3.4 Change the folder where templates are located***

#### **3.4.1 Change the design of the shop**

You can create as many folders you want in "catalog/includes/sts\_templates/" folder and place templates in it. The choice of the folder in use for the shop is made in the admin section, in the "Default" module. You can use this feature to switch between several designs.

By default the template folder is 'test' and it contains only one template. The folder 'full' contains additional templates for the manufacturer page, some categories and a content template for a product. The full folder also contains the new "boxes" folder. This folder is used to create you custom infobox templates. More on infobox templates in section 6.

Use the 'full' folder when you know more about STS and want to try some of the extra possibilities.

#### **3.4.2 Try a new design only for the administrator**

It is possible to change the template folder in use from the URL. This can be useful to test a new design without disturbing the visitors.

By entering the following URL, the templates from folder "mysite" will be used: [www.yourshop.com/index.php?sts\\_template=mysite](http://www.yourshop.com/index.php?sts_template=mysite) .

### ***3.5 Debug mode***

In the admin, select the STS module called "default". You can define the password to be used to access the debug function here. By default the password is "debug".

To see the debug output of index.php, write the URL

[www.yourshop.com/index.php?sts\\_debug=debug](http://www.yourshop.com/index.php?sts_debug=debug). You will then see your page using the corresponding STS template as well as all debug information (version info, template in use, module in use, files included + all placeholders).

If STS is not selected in the default module and you use the debug option, you will be the only one to see the template version of the page.

### ***3.6 Create a template for a specific page***

You can use multiple template files based on the name of the script being executed. You need to:

- Put your custom template files in the template folder using the name format:  
"scriptname.php.html"

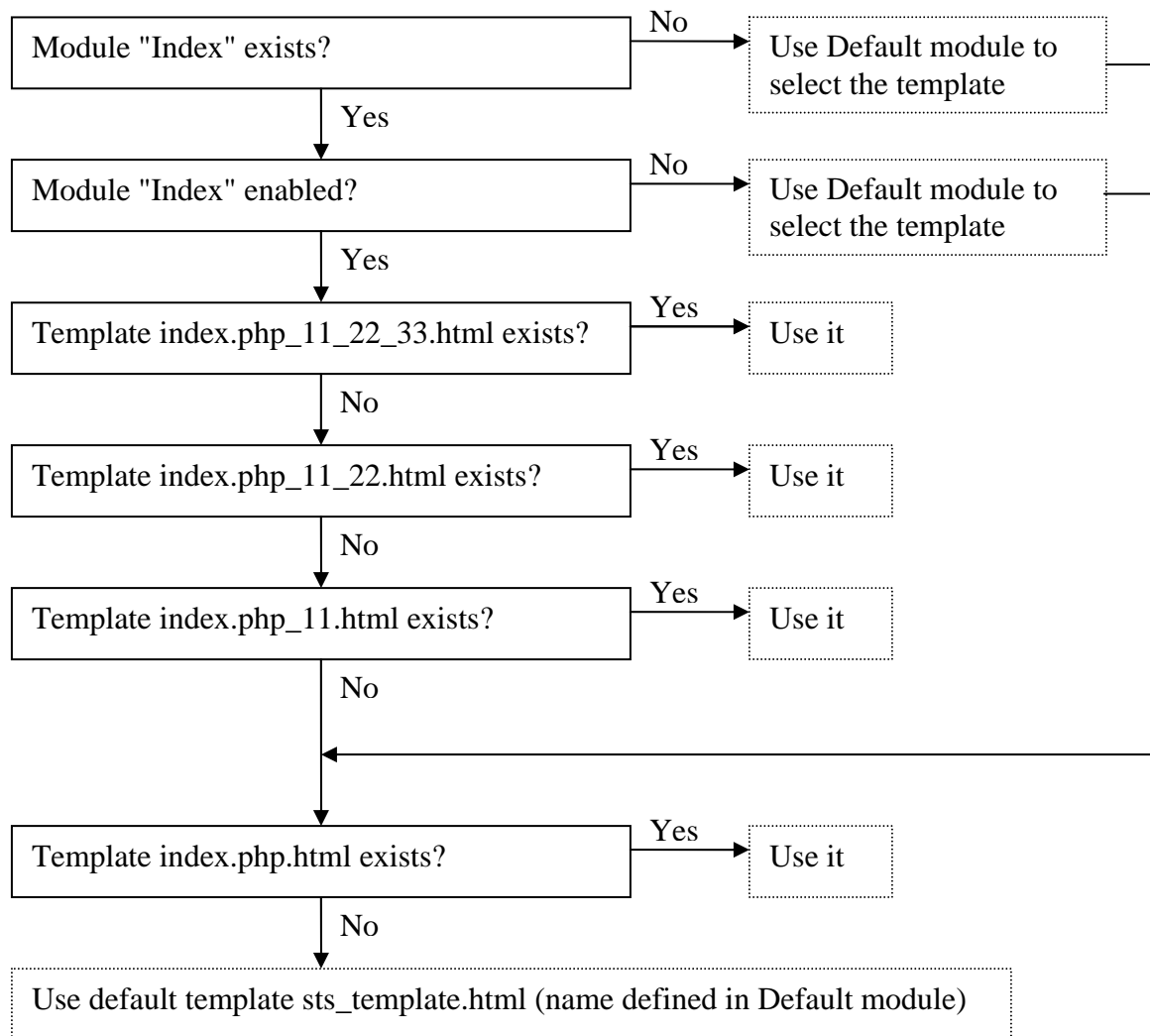
- For example, a custom template file for "index.php" would be:  
“/catalog/includes/sts\_templates/test/index.php.html”
- If a custom template file is not defined for the current script, it will use the default template file instead (defined in the "Default" module).

### 3.7 Create templates for specific categories

You can use multiple template files that will change according to the category selected in index.php. The logic to find the template is made in the "Index" module, so it must be enabled from the admin side.

You can now create custom templates for any or each category. If a template for a category does not exist, the "Index" module will look for the parent categories templates.

Example with the following category: index.php?cPath=11\_22\_33

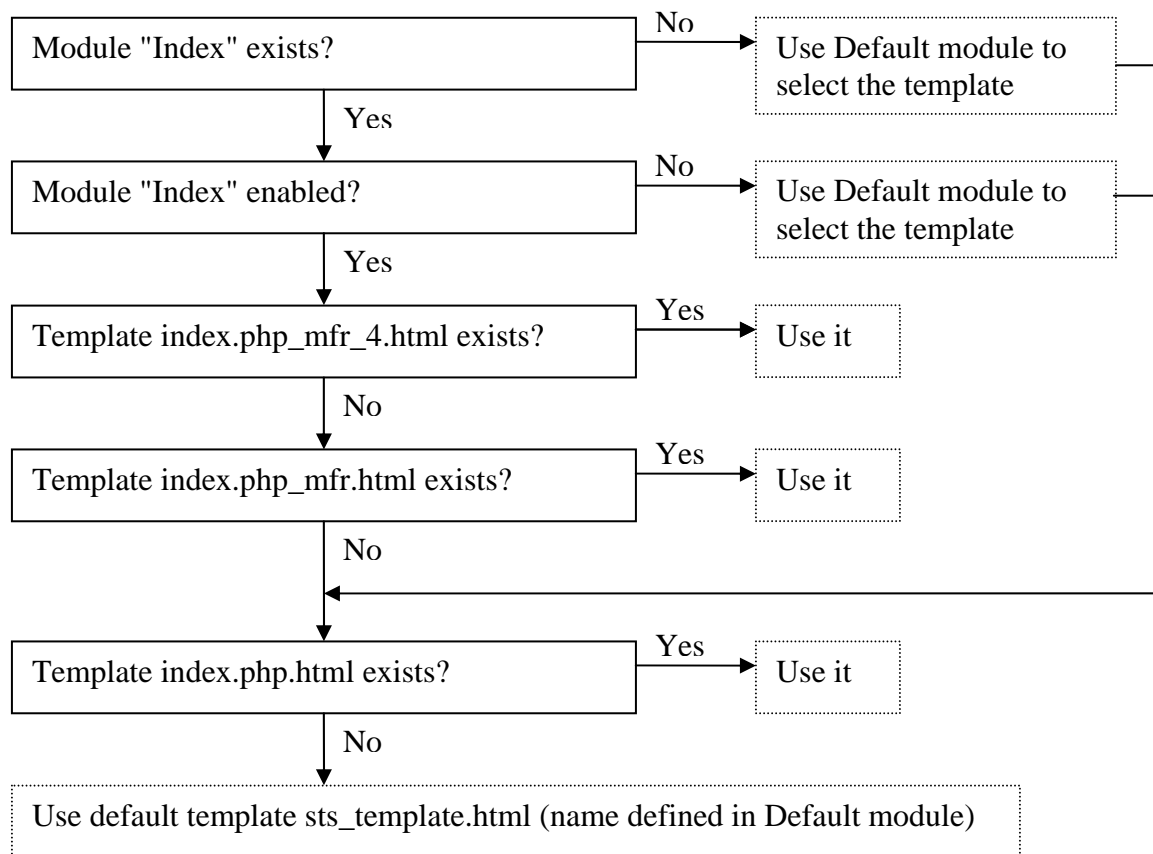


When configuring the "Index" module, you have to give the list of files to include in order to create all the needed variables, in the same way as for the "Default" module. See chapter 3.2.2 for more details.

### 3.8 Create templates for manufacturers

Since v1.0.2 of the index module, you can use multiple template files that will change according to the manufacturer selected in index.php. The logic to find the template is made in the "Index" module, so it must be enabled from the admin side.

You can now create a separate template for each manufacturer, or use the same template for all manufacturers. The following chart shows the decision process when the URL is `index.php?manufacturers_id=4`:



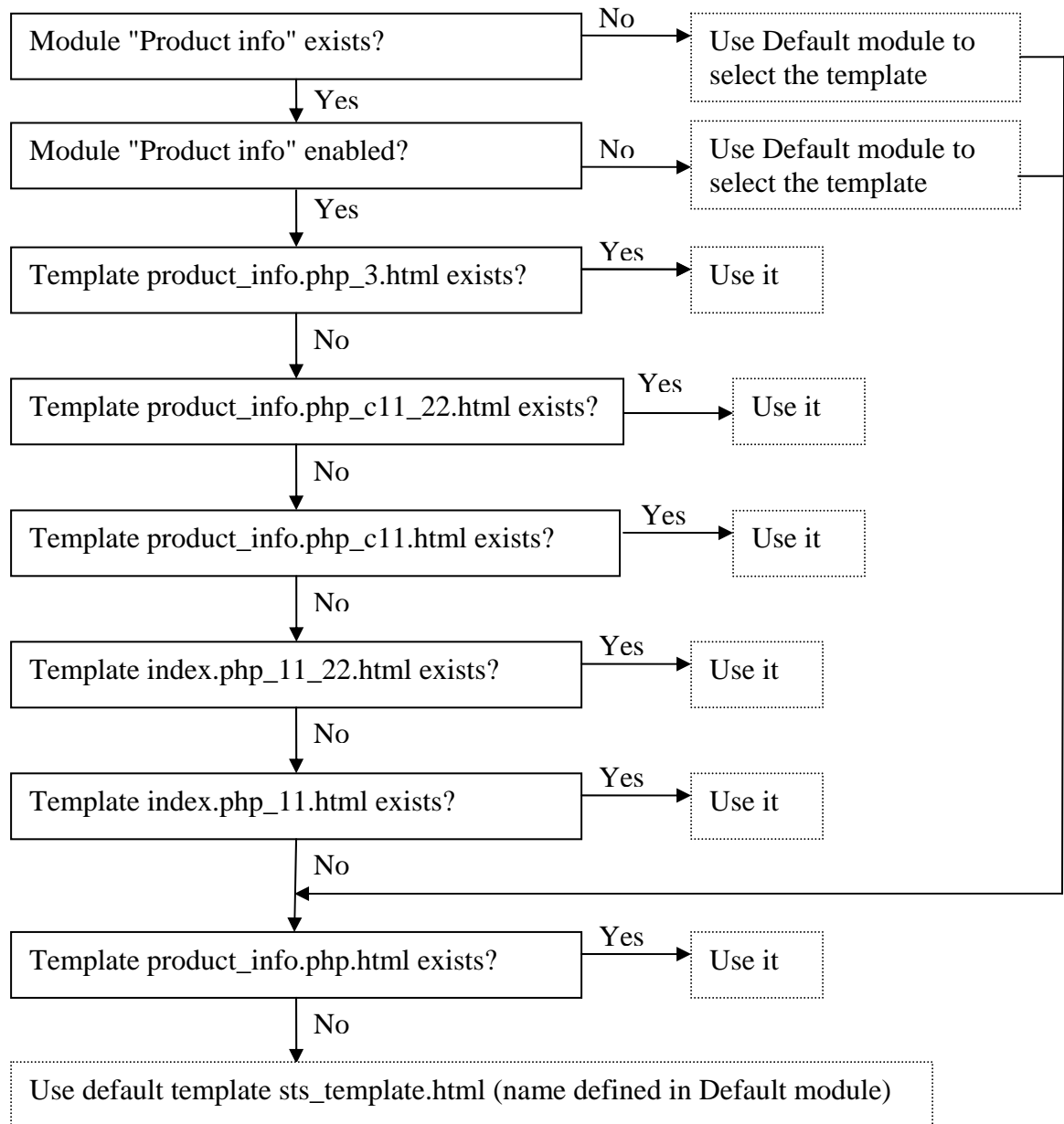
When configuring the "Index" module, you have to give the list of files to include in order creating all the needed variables, in the same way as for the "Default" module. See chapter 3.2.2 for more details.

### 3.9 Create templates for specific products

#### 3.9.1 Template for header, footer and columns

You can use multiple template files that will change according to the product selected in product\_info.php. The logic to find the template is made in the "Product info" module, so it must be enabled from the admin side.

Example with product ID 3 in category 11\_22 (and module v1.0.5):



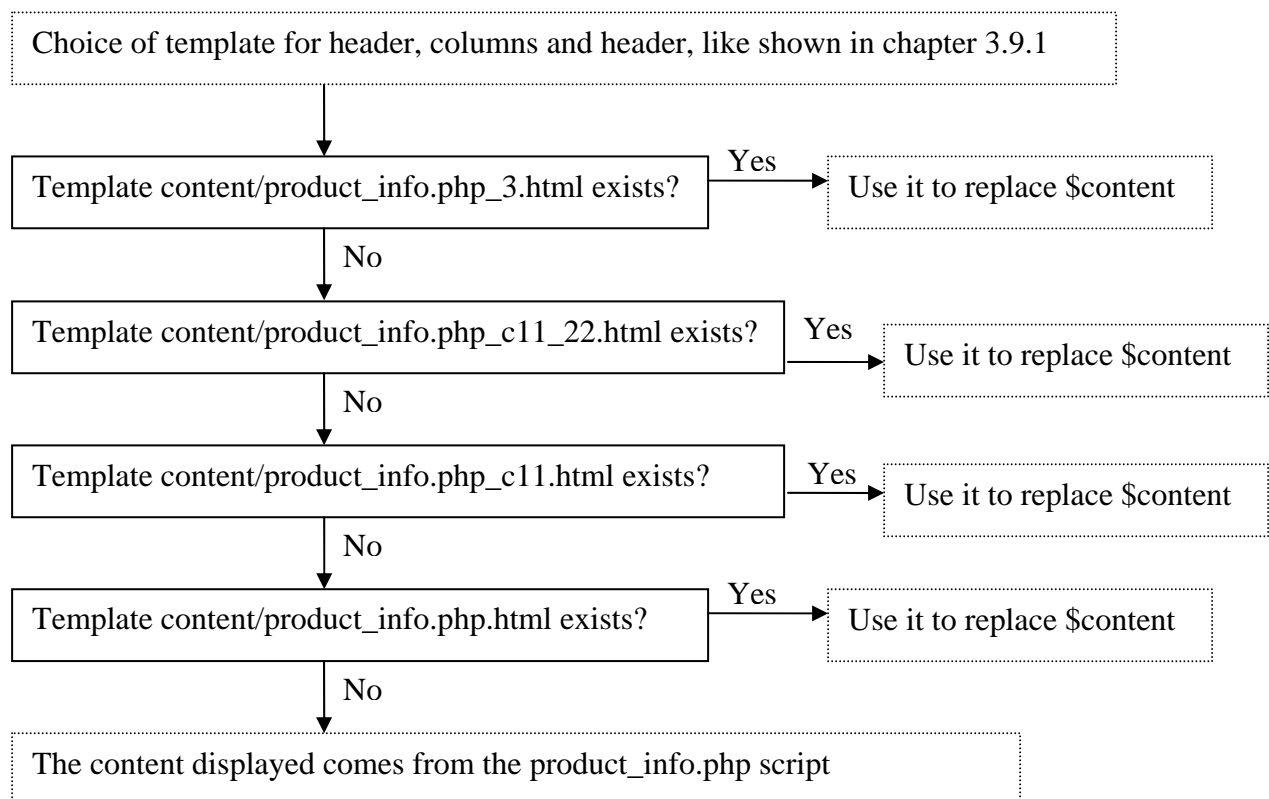
In all these templates, the main content of the page is created by using the `$content` placeholder, which will be replaced by the result of the `product_info.php` script. If you want to create a custom display of the main content for some or all of your products, you have to create a **content template**. See the next chapter for more details.

When configuring the "Product info" module, you have to give the list of files to include in order to create all the needed variables ("Files for normal template" parameter), in the same way as for the "Default" module. See chapter 3.2.2 for more details.

### 3.9.2 Template for the content of the page

To change the position of the content elements of the `product_info` page, you have to create a content template. The folder to put these templates is named "content" and is located inside the folder having all of the other STS templates. For example: `includes/sts_templates/full/content`.

The module "Product info" must be installed and enabled for this system to work. Here is how the template choices are made for product ID 3 in category 11\_22:



When configuring the "Product info" module, you have to give the list of files to include in order creating all the needed variables ("Files for content template"

parameter) in case a content template is used, in the same way as for the "Default" module. See chapter 3.2.2 for more details.

An example of the content template is provided with the contribution and is located here: "includes/sts\_templates/full/content/product\_info.php\_26.html."

This product uses the header/columns/footer template of its category (includes/sts\_templates/test/index.php\_1\_9.html).

If you don't want to use a content template to place the product info elements (e.g. like in STS v2 or v3), you can use the compatibility mode: see chapter 3.9.4.

### 3.9.3 Why a content template?

- More flexibility for design: you can have one content template for all of your products, where you choose the location of price, image and other stuff, while having the header and columns changing according to the category the product belong to. Without content template, you would have to create a separate template for each product to obtain the same result!
- Run the code for the placeholders of the content only when a content template exists. In older versions of STS, this code was processed for all products even if they did not need it which would result in very slow parse times.
- The ability to cache a part of the content template.

NOTE: If you prefer having one template for the whole page, like in previous versions of STS, you can enable the compatibility mode of the product\_info.php module. See next chapter.

### 3.9.4 STS 3 compatibility mode

You can enable this mode from the admin of the product info module. In this mode, when the user is on the product info page, an extra file containing the creation of variables specific to the product info templates will be processed, even if there is no product\_info template (nor general nor specific to product ID). In this way you are sure that your product info templates made for STS v2 or v3 will get all the variables built. But be aware that the parse time of products that use the default template will grow, as the variables will all be processed (same behavior as in STS v2 and v3).

## 4 List of existing placeholders

### 4.1 General placeholders

Here are the tags that are supported by default in STS 4.3 ("by default" means without any extra modules installed).

These tags are created in the file "catalog/includes/modules/sts\_inc/general.php" and are available for all scripts:

Placeholder	Meaning
\$date	The current date
\$sid	The string for the Session ID in the format "SessionIdVarName=WhateverTheSessionIdIs"
\$cataloglogo	Shop logo with link to index.php. Before v4.3: oscommerce.gif in catalog/images folder. Since v4.3: header_logo.gif, in template folder/images/english (or other language).
\$urlcataloglogo	The URL used by the \$catalog logo
\$myaccountlogo	The MyAccount graphic and link to account.php . Before v4.3: header_account.gif in catalog/images folder. Since v4.3: header_account.gif, in template folder/images/english (or other language)
\$urlmyaccountlogo	Deprecated in v4.3, use \$urlmyaccount instead
\$cartlogo	The shopping cart graphic and link to shopping_cart.php. Before v4.3: header_cart.gif in catalog/images folder linking to shopping_cart.php (without SSL). Since v4.3: header_cart.gif, in template folder/images/english (or other language), linking to the same page (with SSL if present).
\$urlcartlogo	Deprecated in v4.3, use \$urlcartcontents instead
\$checkoutlogo	The Checkout graphic and link to checkout_shipping.php. Before v4.3: header_checkout.gif in catalog/images folder. Since v4.3: header_checkout.gif, in template folder/images/english (or other language).
\$urlcheckoutlogo	Deprecated in v4.3, use \$urlcheckout instead.
\$contactlogo	Added in v4.3: image header_contact_us.gif, located in template folder/images/english (or other language), and linking to contact_us.php .
\$loginofflogo	Added in v4.3. If customer is not logged: displays header_login.gif in template folder/images/english (or other language), linking to login.php. If customer is logged: displays header_logoff.gif in template folder/images/english (or other language), linking to logoff.php
\$breadcrumbs	The "Top > Catalog > whatever..." breadcrumbs text and links
\$myaccount	The text version of "My Account" and link to account.php.
\$urlmyaccount	The URL used by the MyAccount function

\$logoff	The Logoff text and link
\$urllogoff	The URL for the LogOff function
\$myaccountlogoff	shows "My Account   Logoff" if logged in or "My Account" if not logged in
\$cartcontents	The text version of the "Cart Contents" function
\$urlcartcontents	The URL of the shopping_cart.php page, using SSL (if available). Buggy before v4.3 (was same as \$cartcontents).
\$checkout	The text version of the "Check Out" function
\$urlcheckout	The URL used by the "Check Out" function
\$templatepath	Deprecated in v4.3, use \$templatedir instead.
\$templatedir	Contains the path to the template folder, without trailing slash.
\$headertags	New in v4.0.4: The meta tags for the <head> section of the html page. By default contains only the <TITLE> meta tag. This variable is automatically inserted into the \$headcontent variable, so <b>you don't need to use it in your template</b> . <u>Note:</u> This variable can be modified by some modules to allow for other meta tags, for example a module for header tags controller modifies this variable to add description and keywords to the pages
\$numrequests	The text "1489 requests since Friday 11 August, 2006"
\$footer_text	The osCommerce copyright information and "Powered By osCommerce" texts.
\$banner_only	The banner, generally displayed at the bottom of the pages.

For compatibility with STS v2 and v3, the variables \$cat\_ and \$urlcat are created in sts\_user\_code.php but are commented out. See chapter 4.6 for details.

## 4.2 Placeholders for infoboxes

Tags for infoboxes are all created in the file  
“catalog/includes/modules/sts\_inc/column\_left.php”:

Placeholder	Meaning
\$categorybox	The Category box
\$manufacturerbox	The Manufacturer box
\$whatsnewbox	The What's New box
\$searchbox	The Search box
\$informationbox	The Information box
\$cartbox	The Shopping Cart box
\$maninfobox	The Manufacturer Info box (empty if no product ID on the page)
\$orderhistorybox	The Order History box (empty if user not logged in)
\$notificationbox	New in v4.0.4: The product notification box (empty if no product ID on the page).

	Do not use this tag together with \$bestsellersbox
\$bestsellersbox_only	New in v4.0.5: The Best Sellers box
\$bestsellersbox	Displays the bestseller box if no product ID on the page or if customer is logged in and has asked to receive notifications for all products. In other cases, will display the notification box. Do not use this tag together with \$notificationbox nor \$bestsellersbox_only
\$specialbox	New in v4.0.4: The Specials box
\$stellafriendbox	New in v4.0: The Tell a Friend box (empty if no product ID on the page)
\$specialfriendbox	Displays the tell a friend box if a product ID is detected and the visitor is not in the tell a friend page, otherwise displays the specials infobox
\$languagebox	The Languages box (empty if on a checkout page)
\$currenciesbox	The Currencies box (empty if on a checkout page)
\$reviewsbox	The Reviews box. Since v4.1, not shown on the product_reviews_info.php page due to a display problem. Infobox and this page are using same variable names and it makes a mess of your template.

### 4.3 Placeholder for main content

The main content of the page (middle of the page) is put in the \$content tag. The content is taken from the PHP script you are running. For example, the \$content for index.php is the piece of code you find in index.php, between the call of column\_left and the call of column\_right. If you want to modify it, you have to modify the script itself.

You can also use templates for the content part only. See example in chapter 3.9.2.

### 4.4 Special placeholder for html header

Important tags to put in the header of the html template:

\$htmlparams	New in v4.0.7: Direction and language of the page text. To be put in the html tag of the template: <html \$htmlparams>
<!--\$headcontent-->	Tag to put in the <head> of your template. It contains the charset of the page, meta tags, base href and javascript taken out of header.php.

	<p>You must use this commented format of \$headcontent to keep the word "\$headcontent" from displaying in your web authoring software and be replaced by the correct value.</p> <p><u>NOTE:</u> Do not put &lt;title&gt;...&lt;/title&gt; tags in your header. It will be added as part of the \$headcontent variable.</p>
<!--\$extracss-->	<p>Optional. Allows placing a link to a stylesheet dynamically. Will be replaced by the content of \$sts-&gt;template['extracss']. The whole html line must be placed in, like &lt;link rel="stylesheet" type="text/css" href="stylesheet.css"&gt;</p> <p>You must use this commented format of \$extracss to keep the word "\$extracss" from displaying in your web authoring software and be replaced by the correct value.</p>

## 4.5 Placeholders for footer

Tags generally put in the footer of the page, created in the file “catalog/includes/modules/sts\_inc/general.php”:

Placeholder	Meaning
\$numrequests	The "XXX requests Since DATE" text
\$footer_text	New in v4.0.5: the footer text, from the constant FOOTER_TEXT_BODY
\$banner_only	New in v4.0.5: the banner, without html table

For compatibility with templates made with STS v2 or v3, the variables \$footer and \$banner are created in sts\_user\_code but are commented out by default. See next chapter.

## 4.6 Placeholder for user created tags

The file “catalog/includes/modules/sts\_inc/sts\_user\_code.php” is made for your own user tags. Put the tags that you create yourself in this file. You can also create a new file in the same folder, create the tags in this new file and configure the STS modules to include this file. The “sts\_user\_code.php” already contains one tag:

NOTE: If you don't need this file, you can remove it from the STS module configuration and save some execution time.

Placeholder	Meaning
\$catmenu	A drop-down list of available categories
\$footer	Includes \$footer_text as well as date and requests counter, inside an html table. To use this variable, uncomment its code. Existing only for compatibility with templates made with STS v2 and v3
\$banner	The banner inside an html table. To use this variable, uncomment its. Existing only for compatibility with templates made with STS v2 and v3
\$cat_CategoryName, \$urlcat_CategoryName	Links to categories using the category name. Available for each language. Example: \$cat_Hardware. To use these variables, uncomment their code.
\$cat_CategoryID, \$urlcat_CategoryID	Links to categories using the category ID. Better use the ID than the name, so changing the category name won't break the link. Example: \$cat_2. To use these variables, uncomment their code.

Note about \$cat and \$urlcat variables: to use them, uncomment their code. Then, if the system does not find any of these variables inside the template, the variables will not be created, thus saving execution time. It is not recommended to use these variables as it is a very slow process!

#### **4.7 Placeholders for product\_info.php content template**

These are the tags you can use when creating product\_info templates. They are created in the file: “includes/modules/sts\_inc/product\_info.php”. If you want to use this feature, you have to enable the product\_info module in admin.

<b>Placeholder</b>	<b>Meaning</b>
\$productid, \$productsid	The ID of the product being displayed. Added in v4.0.6 for compatibility with STS v3
\$startform	Required as the first item in the content area. This starts the HTML form needed for adding things to your cart
\$endform	This should be the last item in the content area. This closes the HTML form needed for adding things to your cart
\$regularprice	The price of the product, does not care of special prices
\$specialprice	The special price, empty if not defined
\$regularpricestrike	Contains "<s>\$regularprice</s>" if a specialprice is set, otherwise contains \$regularprice (not struck).
\$productname	The product name
\$productmodel	The product's model number
\$productdesc	The product description
\$imagesmall	The product image scaled down to the thumbnail size specified in your /admin/ tools. Since v4.0.7 it is empty if there is no product image. Use \$product_popup if you want to have the popup option.
\$imagelarge	The product image with no resizing (actual size of image). Since v4.0.7 it is empty if there is no product image.
\$product_popup	New in v4.0.7: The Javascript with small image and link to popup window. Empty if no product image.
\$optionheader	The "Available Options:" text above the options, will change with the language. Blank if no options (product attributes)
\$optionnames	A list of Option (Product Attribute) names, separated by line breaks (" "). Blank if no options
\$optionchoices	A list of Option (Product Attributes) choices as drop-down boxes, separated by line breaks (" "). Blank if no options
<b>NOTE:</b> As shown in the “/catalog/includes/sts_templates/test/product_info.php_26.html” file, it is recommended that you use an HTML table to layout the options (Product Attributes). In the first column have \$optionnames and in the second column have \$optionchoices. This will automatically keep the correct names next to the correct choices. Use a table without a border.	
\$moreinfo	The language-sensitive value of TEXT_MORE_INFORMATION (for

	linking to a more detailed page)
\$moreinfourl	The URL of the page where more information can be found
\$productdatelabel	Will contain either the Date Added or the Date Available messages, language specific
\$productdate	The date it will be available or the date it was added
\$reviews	The text with the number of reviews for the item. Empty if no reviews are available
\$reviewsbutton	The image for the reviews button
\$reviewsurl	The URL for the product's reviews
\$addtocartbutton	The button to send the product into the cart
\$alsopurchased	The box with products also purchased *added in 4.0.6

## 4.8 Placeholders for product\_info.php template in compatibility mode

If you want to create templates for the product info page, where you can place prices, images and texts where you want, it is **HIGHLY RECOMMENDED** to use the content template method. If you don't want to use a content template for your products but rather use a single template which includes the columns, header, footer, product info placeholders (but no \$content), you would need to enable the compatibility mode in the product info module. When this mode is enabled, the same placeholders are available as in the precedent chapter, plus the \$sysmsgs. This placeholder must be placed where you want the system messages to be displayed.

## 5 Tips and tricks

### 5.1 Absolute URLs

You need to think carefully about the links you make when working with osCommerce. The same applies to images, because there are quite a few potential pitfalls to avoid in how you link to them when designing templates and new pages for your shop.

Basically, you will be in the clear as long as your image links are relative. Make sure you link to *'images/mylogo.gif'* and **never** to *'http://www.mysite.com/catalog/images/mylogo.gif'*.

The second method would work initially, but will then lead to errors when you create a secure area of the site to handle credit card transactions. In such situations, even images must be transmitted in an encrypted form to avoid security alerts. By specifying that you're using the 'http' protocol and not the secure 'https' protocol, it causes a problem. If you think the path is correct but images still don't appear, check that you've actually uploaded them to the server in the right directory – this is a surprisingly easy mistake to make.

## 5.2 Save some execution time

### 5.2.1 Change placeholder names

All variables created for the templates are placed in an array that needs to be sorted before replacing the placeholders, so placeholders are replaced starting from longest name to shortest name. Why? Look at these two placeholders: \$prod contains the text "Hello", \$product\_image contains a link to an image. If array is not sorted, the risk is that \$product\_image becomes \$Hellouct\_image during the replacement process, breaking the template.

By adding a stop char at the end of placeholders, like \$prod\$ instead of \$prod, the sorting of the big array containing all the placeholders variables is not needed anymore.

To do so, you have to open your template, add the same end char to all placeholders, in our example "\$", then tell the system to use this end char. This is done by defining a constant in the template: add this code at the beginning of your template to indicate you are using an end char:

```
<?php DEFINE ('STS_END_CHAR', '$'); ?>
```

For content templates, add this instead of the above code:

```
<?php DEFINE ('STS_CONTENT_END_CHAR', '$'); ?>
```

The system will detect that an end char is used and will not sort the big array. If you already have several templates, this method allows you to modify them one by one.

**NOTE: You can use the symbol, \$ or #, to get the same result. Some of the included STS templates already use the # symbol and have already placed the above code at the beginning of the HTML template for you.**

There was a bug before v4.0.6: the placeholders were correctly replaced but the array was always sorted.

### 5.2.2 Unused code

You can check in “includes/modules/sts\_inc/sts\_user\_code.php” and “general.php” (in the same folder) for variables that you never use and remove them from the code.

### 5.3 Create your own buttons (optional)

Want to really make your STS Template really stand out? Change the standard buttons throughout your store to match your design. To help you out with this, I have included a complete button index for you to work from. Once you have edited these files, save them in your “*catalog/includes/languages/english/images/buttons*”. Note: If you use a language other than “English”, you will have to save your buttons in those corresponding language folders as well.

#### BUTTON INDEX

The list below shows the buttons you need to create to make a full custom set for osCommerce, and specifies their filenames as well:

- button\_back.gif .....Back
- button\_buy\_now.gif.....Buy Now
- button\_change\_address.gif.....Change Address
- button\_checkout.gif .....Checkout
- button\_confirm\_order.gif .....Confirm Order
- button\_continue.gif .....Continue
- button\_continue\_shopping.gif .....Continue Shopping
- button\_delete.gif .....Delete
- button\_edit\_account.gif.....Edit Account
- button\_history.gif .....Order History
- button\_in\_cart.gif .....Add To Cart
- button\_login.gif.....Login
- button\_notifications.gif .....Notifications
- button\_remove\_notifications.gif ...Remove Notifications
- button\_reviews.gif .....Reviews
- button\_search.gif .....Search
- button\_shipping\_options.gif .....Shipping Options
- button\_update.gif .....Update
- button\_update\_cart.gif .....Update Cart
- button\_add\_address.gif.....Add Address
- button\_write\_review.gif .....Write Review
- button\_address\_book.gif.....Address Book
- Small\_delete.gif .....delete (half normal size)
- Small\_view.gif.....view (half normal size)
- Small\_edit.gif.....edit (half normal size)

## 5.4 Create your own images (optional)

In the last section, I explained how to create a custom look for your osCommerce site just by creating your own images. You can do the same with all of the default osCommerce images. There are various graphics which feature on particular pages of the standard osCommerce site, which you might want to replace or remove in the interests of making the site design cohesive and have a feeling of being unique rather than off-the-shelf.

In the table shown here, I've summarized the filenames and descriptions of many such graphics, to aid you with designing replacements for them. Should you wish to remove the graphics from use entirely, that is best done by accessing the pages which they are used on, and deleting the relevant image tags. Don't just delete the offending image files themselves without editing the code, because you'll end up with an even worse situation of endless missing image symbols across your site.

You can easily replace the graphics with your own by "saving" over the default images with your custom graphic of the same file name and folder as the original

FILE NAME	DESCRIPTION	LOCATION
images/arrow_south_east.gif	White arrow pointing down and right	Various locations
images/arrow_east_south.gif	White arrow pointing right and down	Various locations
images/arrow_down.gif	Arrow pointing down	Various locations
images/table_background_default.gif	Person at cashdesk	Top right of many pages, alongside headline
images/table_background_payment.gif	Credit cards	Checkout page
images/table_background_confirmation.gif	Receipt emerging from machine	Order confirmation page
images/table_background_man_on_board.gif	Suited man on surfboard (huge)	Final order confirmation page
images/checkout_bullet.gif	Orange bullet point	Marks stages of order process
images/infobox/arrow_right.gif	White arrow pointing right	Top right corner of infoboxes, always linked to other page
includes/languages/english/images/icon.gif	Union Jack flag	Languages infobox (NB: also change other flags)
includes/languages/english/images/buttons/button_tell_a_friend.gif	Yellow envelope	Tell A Friend infobox (NB: also change other language versions)
includes/languages/english/images/buttons/button_quick_find.gif	Cartoon-style magnifying glass	Quick Search infobox (NB: also change other language versions)
images/stars_1.gif to images/stars_5.gif	5 images with varying quantity of stars	Various review-related locations

images/table_background_reviews_new.gif	People on a book	Top right corner of review-related pages
images/table_background_browse.gif	Wooden cartoon signpost	Top right corner of search-related pages
images/table_background_cart.gif	Shopping bags	Top right corner of shopping cart pages
images/table_background_login.gif	Man walking into window	Top right corner of login pages
images/table_background_account.gif	Pen writing in book	Top right corner of account page
images/account_personal.gif	Manilla folder	My Account page
images/account_orders.gif	Boxes with clipboard	My Account page
images/account_notifications.gif	Cartoon envelope	My Account page

## 5.5 Replace Page Heading Title with the name of Categories or Subcategories (optional)

Get rid of "Let's See What We Have Here" for Subcategories and "Categories" for top-categories. You can dynamically replace them with the name of the categories or subcategories name. This is a very cool hack.

In the “/catalog/index.php” file, there are three lines of code such as:

```
<?php echo HEADING_TITLE; ?>
```

*FOR CATEGORIES, find the first one:*

```
<?php echo HEADING_TITLE; ?>
```

*Replace with:*

```
<?php echo $category['categories_name'] ?>
```

*FOR SUBCATEGORIES, find the second one:*

```
<?php echo HEADING_TITLE; ?>
```

Replace with:

```
<?php if (isset($HTTP_GET_VARS['manufacturers_id'])) {
    $category_query = tep_db_query("select manufacturers_name from "
    . TABLE_MANUFACTURERS . " where manufacturers_id = " .
    (int)$HTTP_GET_VARS['manufacturers_id'] . " ");
    $category = tep_db_fetch_array($category_query);
    if ($category['manufacturers_name'] != "") {echo
    $category['manufacturers_name'];} else { echo HEADING_TITLE;}
    } else {
        $category_query = tep_db_query("select cd.categories_name from "
        . TABLE_CATEGORIES . " c, "
        . TABLE_CATEGORIES_DESCRIPTION . " cd where c.categories_id = "
        . (int)$current_category_id . " and cd.categories_id = "
        . (int)$current_category_id . " and cd.language_id = " . (int)$languages_id
        . " ");
        $category = tep_db_fetch_array($category_query);
        if ($category['categories_name'] != "") {echo
        $category['categories_name'];} else { echo HEADING_TITLE;}
    }
?>
```

There is nothing to modify with the third line of code **<?php echo HEADING\_TITLE; ?>** so just leave it be.

## 5.6 Create your own STS modules (optional)

OsCommerce uses modules such as the shipping or payment modules to allow you to easily “turn on/off” options for your store in the admin panel. This contribution comes with three STS modules that include the Header Tags Controller module and the Infobox Template module integrated. What could be easier?

### 5.6.1 Why create STS modules?

Each module takes care of selecting the template corresponding to a particular script.

You may want to create a different template when displaying the new products page or create templates to display articles after having installed a contribution.

### 5.6.2 How?

The easiest way is to copy an existing module and rename it to the name of the script you want to use as templates. If your script is articles.php, your module would be “includes/modules/sts/sts\_articles.php”. In this module, you would have to modify the “find\_template” function to select the template accordingly to the example for the article ID to be displayed.

If you want to select a template according to the manufacturer page, you have to modify the module “includes/modules/sts/sts\_index.php” to look for the manufacturer ID.

## **5.7 Create your own placeholder file to include (optional)**

### **5.7.1 Why?**

Placeholder files contain code to create the variables used to replace the placeholders of the templates. This code can be put in the sts\_user\_code.php file, but placed in a separate file it allows, via the modules configuration, to limit the execution of the code only to the modules that need these new variables.

Example: You want to create new variables for a contribution that displays several images on the product info page (hint hint). If you put it in sts\_user\_code.php, all pages will run this code even it is not used. If you place it in a separate file and configure only the product info module to use it, the code will run only when needed.

It is also easier to share with fellow STS users (hint hint).

Example: Somebody has created a file with variables for the header tags contribution. To use it, you only need to copy it and to configure the modules that will use it, via the admin section of your store. In fact, that was what was done in creating the bundled HTC in this contribution. What could be easier?

### **5.7.2 How?**

Put the code in a file, put the file in the “includes/modules/sts\_inc/” folder and configure the modules that will use this file.

## 6 Infobox templates

### 6.1 Description

Since v4.3, the infobox template system is integrated inside the STS package and is not a package to download separately. It can be turned on and off from the STS default module.

An Infobox template contains only plain HTML code, without PHP includes. You can edit them in visual HTML editors, like Dreamweaver, FrontPage, etc.

An Infobox template consists of one or two files:

- Infobox Header (optional):  
/includes/sts\_templates/TEMPLATE\_NAME/boxes/infobox\_header.php.html
- Infobox Content:  
/includes/sts\_templates/TEMPLATE\_NAME/boxes/infobox.php.html

The above two files are pure html files with only 3 labels (placeholders):

**\$headertext** - title of infobox

**\$right\_arrow** - right arrow in the corner of the infobox header with hyperlink to script page. (ex: specials.php)

**\$content** - content inside the infobox.

You can create any html-code for your infoboxes, simply by inserting in it the three labels above.

It is possible to create templates for individual infoboxes. For example, if you want the “Information” infobox to be different from the others, create a separate template for this infobox by creating the following files:

- /includes/sts\_templates/TEMPLATE\_NAME/boxes/infobox\_information\_header.php.html (optional)
- /includes/sts\_templates/TEMPLATE\_NAME/boxes/infobox\_information.php.html

Include the name of the infobox template in the filename in the following format:

- infobox\_infoboxname.php.html
- You can find the list of infoboxes used in your store in the “catalog/includes/boxes” or in “catalog/includes/modules/sts\_inc/sts\_column\_left.php”

Creative ideas for your infobox templates:

1. Use your stylesheet to set the background to transparent to better match your customized template page

2. Look for the following code in your stylesheet.css file:

```

/*This is the INFOBOX Background Style, one shade darker than the
Infobox Header color*/
.infoBox {
    background: #0000FF;
}

/*This is the INFOBOX Contents Style, one shade lighter than the Infobox
Header color*/
.infoBoxContents {
    background: #FFFFFF;
    font-family: Verdana, Arial, sans-serif;
    font-size: 10px;
}

/*This is the INFOBOX Notice Style, a pale salmon color*/
.infoBoxNotice {
    background: #FF8E90;
}

/*This is the INFOBOX Notice Contents Style, a very pale salmon color*/
.infoBoxNoticeContents {
    background: #FFFFFF;
    font-family: Verdana, Arial, sans-serif;
    font-size: 10px;
}

/*This is the INFOBOX Heading Style, SAME AS HEADER AND
FOOTER GRAY BARS*/
TD.infoBoxHeading {
    font-family: Verdana, Arial, sans-serif;
    font-size: 10px;
    font-weight: bold;
    background: #FFFFFF;
    color: #ABDA4E;
}

TD.infoBox, SPAN.infoBox {
    font-family: Verdana, Arial, sans-serif;
    font-size: 10px;
}

```

3. Just by changing the above, you can make the borders, backgrounds and more completely dependent upon your design. In other words, it is now a lot easier to integrate STS into a store bought HTML template. You can

also take advantage of newer CSS dynamic boxes. Have fun, be creative.

4. The default tables show a gray background around each infobox. This can easily be removed by editing the table and cell within the infobox template. You may want to replace the corner images with transparent gifs. These images are located in catalog/images/infobox/.

## 6.2 Test

Activate template 'full' in admin:

In Admin/Modules/STS/Default module, set "Template folder" = full and activate infobox templates with "Use template for infoboxes" = true.

The infoboxes are now using the templates defined in  
/includes/sts\_templates/full/boxes/

---

## 7 Support Forum

A support forum thread has been created for STS v4.x:

<http://forums.osCommerce.com/index.php?showtopic=204381&st=0>

It is a good idea to search the forum for any questions you may have before actually posting a new topic.

If you prefer to use Google (www.google.com), you can use the following Google search string to facilitate your search:

["sts4" "double header" site:forums.osCommerce.com](#)

The example above will first search in the “*STS4*” osCommerce forum, and then search for any threads regarding the “*double header*” issue.  
What could be easier?

Here is another example:

[text edits site:www.osCommerce.info/](#)

The example above will search the osCommerce Knowledge Base for “*text edits*”. This is great for finding what pages to edit to get rid of the generic text of the default osCommerce store. Ok, I had better stop on this one...getting a little carried away. ☺

---

## 7.1 STS 060817 module patch:

For those who are using osCommerce version 060817:

You may be experiencing difficulty in activating the STS module in the osCommerce admin panel.

It appears this is a general problem with the compatibility.php file in the 060817 update. Here is a work around: Add the following code

```
reset($ar);
```

at the end of the do\_magic\_quotes\_gpc routine.

It will end up looking like this:

```
function do_magic_quotes_gpc(&$ar) {
    if (!is_array($ar)) return false;
    while (list($key, $value) = each($ar)) {
        if (is_array($ar[$key])) {
            do_magic_quotes_gpc($ar[$key]);
        } else {
            $ar[$key] = addslashes($value);
        }
    }
    reset($ar);
}
```

This is not an STS issue, but rather a bug in the latest osC update. This will also correct other module problems such as payment and shipping module activations.

STS v4.3 with HTC v2.5.9 document by Bill Kellum of Sounds Good Productions (<http://www.soundsgoodpro.com>) STS v4.3 document by Christophe Büchi aka Rigadin ([rigadin@osc-help.net](mailto:rigadin@osc-help.net)). It is based on v4.2 document that was edited for new content and errors as of August 8, 2006 by Bill Kellum ([info@billykellum.com](mailto:info@billykellum.com)) of Sounds Good Productions.

Author of “Infobox Templates for STS”:

Grigory Kopanev ([gtkopanev@mail.ru](mailto:gtkopanev@mail.ru)), modified by Christophe Büchi aka Rigadin ([rigadin@osc-help.net](mailto:rigadin@osc-help.net)) for STS v4.3, modified by Bill Kellum ([info@soundsgoodpro.com](mailto:info@soundsgoodpro.com)) for use with STS v4.2 and STS v4.3 with HTC v2.5.9.