STS 4.6

Updated as of February 14, 2010 by Bill Kellum “bkellum”

Table of contents

1 STS Tutorial

I have a STS Tutorial available on my web site that will guide you every step of the way in creating a custom STS Template from the beginning to the end. You can get it here: http://soundsgoodpro.com/eshop/index.php?cPath=26

With that said, STSv4.6 is easy enough on its own without the need of a tutorial just by utilizing this free User Manual (what you are reading now). I strongly suggest that you go through each of the sample templates to get a grip on how STS functions.

2 Basics about STS

2.1 *What is it?*

STS is an easy to install template system for osCommerce Online Merchant versions 2.2MS2, RC1, RC2 and RC2a. The graphical part of the shop can be designed using a standard html editor like Dreamweaver, Expression Web or FrontPage.

If this contribution helps you a lot and you want to help in continuing the development, any donation amount to the following PayPal account would be a great motivation: billkellum@soundsgoodpro.com
Or donate via my web site at:
http://soundsgoodpro.com/eshop/product_info.php?products_id=63

Thank you very much and I hope you enjoy this contribution!
Bill Kellum "bkellum"

2.2 *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 core 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 or a specific product and all infoboxes (see chapter 4.6 for more details on multiple templates).

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/**blank**/sts_blank_template.html".

3 Installation

Refer to separate document for the installation (included in this download located at ***docs/install.html***). After installation, STS **is not enabled by default** and your shop looks the same as before. Continue to read from here once you have completed the installation to enable STS just as you would a stock payment or shipping module in the admin.

4 How to use STS:

4.1 Simple, One Page Template

Follow these steps if all you are wanting is a single template to be used for all of the pages in your shop:

From the admin panel in your osCommerce store, go to Modules and select STS. Once you opened the STS Module, click on the Default Module. You will need to install this module by clicking on the Install button. Once installed, you will need to enable the module by clicking on the “True” radio button.

Select the template folder “single” from the template folder drop down menu. The look of your shop will now change to the look of the template that is located in:

- /catalog/includes/sts_templates/**single**/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 4.3 if you want to change the name of the template.
- See chapter 4.4 if you want to change the folder where the templates are located.
- **See chapter 4.6 for using STS to create more than a single template by creating templates for specific pages.**
- See chapter 5 for a list of available placeholders.
- See chapter 10 for a brief description of each sample template set.

Even though this is an easy feature in STS, it is highly recommended to take advantage of multiple template pages that allows you to have a truly custom look and feel for your osCommerce shop. See chapter 4.6 for more details.

4.2 Add new boxes or template variables

4.2.1 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 the correct code syntax is provided in the file.

- See chapter 6.5 for another way to create new variables.

4.2.2 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 the STS Default module: add this new file name after sts_use_code.php in the “**Files for a Normal Template**” field.

  There can be several names defined here. Each must be separated by a semicolon (example: **sts_user_code.php;header_tags.php**).

4.3 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. YOU MUST HAVE A DEFAULT TEMPLATE in order for STS to function properly.

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

- /catalog/includes/sts_templates/single/**sts_template.html**

The name of the default template in use can be changed by editing the “STS Default" module from the admin panel. Most would simply keep the default template named sts_template.html. You would edit this template to meet your needs or create one on your own and name it sts_template.html.

4.4 Change the folder where templates are located

4.4.1 Change the design of the shop

You can create as many folders you want in the Base Folder “catalog/includes/**sts_templates**/” and place templates in it. The choice of the folder in use for the shop is made in the admin section, in the "**STS Default**" module, by using the convenient “Template Folder” drop down menu. You can use this feature to switch between several designs in an instant.

Note: You must upload any new template folders to the “**Base Folder**” before they will be made available in the drop down menu. For example: If you want to make your new “**christmas**” folder available in the drop down menu, you would need to upload this new folder using FTP (file transfer protocol) to the STS Base Folder which is the following: “catalog/includes/**sts_templates**/”. Once you have uploaded your new template set, it would be “catalog/includes/sts_templates/**christmas**/”.

By default, the template folder is 'single' and it contains only one template. The folder 'multiple' contains additional templates for the manufacturer page, some categories and a content template for a product. Use the 'multiple' folder when you know more about STS and want to try some of the extra possibilities. Of course, you are not limited to just these two template folders. You can have as many template folders as you like. Suggestions: Holiday templates, Special Sales templates, etc. The “single” and “multiple” templates folders are provided only as a sample for you to design your own.

NEW in STSv4.6 you will see a few other open source template sets available for use as you see fit. Feel free to modify any of these template sets to meet your needs. See chapter 10 for a brief description of each sample template set.

4.4.2 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.

Suppose you wanted to try out a new template folder “mytemplates” without the whole world seeing it as well. This is simple to do.
By entering the following URL, the templates from folder "mytemplates" will be used:
www.yourshop.com/index.php?sts_template=**mytemplates**

4.4.3 Move templates out of includes/sts_templates

Since v4.4, the folder "includes/sts_templates/", where all template folders are located, can be changed from the admin side, under the STS Default module. The parameter you enter should end with a / and is relative to your catalog files. So if

your store is located in "www.mystore.com" and you want to change this parameter to be "templates", you would enter the following:
templates/
Using the “template” folder example above, and you have a template folder named "multiple", your templates will be located in "www.mystore.com/**templates**/multiple/".

If you decide to change the name of the default template Base Folder, keep this in mind when uploading new template sets from third party sources as the installation path would need to be modified to your new Base Folder name and location.

4.5 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**.

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.

4.6 Create a template for a specific page

This is what most people install STS for in the first place. STS allows you to create as many specific templates for specific pages in your shop. You can use multiple template files based on the name of the script being executed. You need to:

- Put your custom template files in your template folder using the name format:
  “**scriptname.php.html**”
- For example, a custom template file for "**index.php**" would be “**index.php.html**” and would be uploaded to (if your template folder name is “multiple”):
  “/catalog/includes/sts_templates/**multiple/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 "STS Default" module).

4.7 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 "STS Index" module, so it must be enabled from the admin side.

The logic to choose the template to use is built in the module. Please refer to the “**module_index.html**” file in the “**docs**” **folder of this download** for more details.

Create templates for manufacturers
Since v1.0.2 of the STS 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 "STS Index" module, so it must be enabled from the admin side.

The logic to choose the template to use is built in the module. Please refer to the "**module_index.html**" file in the "**docs**" **folder of this download** for more details.

4.8 Create templates for specific products

4.8.1 Main 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 "STS Product info" module, so it must be enabled from the admin side.

The logic to choose the template to use is built in the module. Please refer to the "**module_product_info.html**" file in the "**docs**" **folder of this download** for more details.

4.8.2 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 template folder having all of the other STS templates. For example: includes/sts_templates/multiple/**content**/.

Note: The module "**STS Product info**" must be installed and enabled in the admin for this system to work.

The logic to choose the template to use is built in the module. Please refer to the "**module_product_info.html**" file in the **"docs" folder of this download** for more details.

4.8.3 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 belongs to. Without a 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.

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 (not recommended -See next chapter).

4.8.4 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 (not 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).

4.9 Create templates for product image popup window

Since STS v4.4, a template can be used to display the product popup image window popup_image.php. Create a template named **popup_image.php.html** and place it in your template folder.

The logic to choose the template to use is built in the module. Please refer to the "**module_popup_image.html**" file in the "**docs" folder of this download** for more details.

4.10 Create content templates for any page

Since v2.1.1 of module sts_default.php (delivered with STS v4.5) you can create content templates for whatever page you want, without creating a new module. This is particularly useful when creating new pages.

The catalog script that is going to use a content template must place the template name in the variable $sts->script->content_template_file, for example:
STS_TEMPLATE_DIR . "content/mypage.php.html";

A variable for a default content template is ready for use in the sts class:
$sts->default_content_template and contains content/sts_template.html

So you can create logic in your script to check if a specific content template exists, and use the default one if not.

In your script you now have to create an array with placeholder names and values, placeholders that you will be used on your content template. The array is the format:
$template_content ['placeholdername'] = 'placeholdervalue';

On your content template you can now use {$placeholdername} (the leading and trailing chars are fixed). This is important to not change the name of the array!
If you use the default template for several pages, you must be sure that all pages create the same placeholders, even empty, so you won't see placeholder names in your browser.

4.11 Replace 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. There are two ways to replace these images.

4.11.1 Replace the original images, all template sets

Default osCommerce images are located in catalog/images folder. Just replace these images with the ones you like. The changes will affect all the template sets you have so it does not give you much flexibility. See the following sections for more flexible options with STS.

In the table shown below is a summary of 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. **Note: The other possibility is to replace the image with a transparent one. There are several transparent image contributions available for free download.**

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

4.11.2 Replace the original images, for one template set

Since v4.4, each template set can use its own set of images. If an image is not found in the template set, the default osCommerce image will be used.

To replace an osCommerce image that is located in catalog/images/, place the custom image with the same name in the images folder of your template folder. For example (if your template folder in use was named “mytemplates”): catalog/includes/sts_templates/**mytemplates/images**/ and STS will use that image in that template set only. This allows you to create images specific to the theme of each of your custom template folders.

4.12 Create your own buttons

Want to really make your STS Template really stand out? Change the standard buttons throughout your store to match the theme of your template. As you will see in section 4.13.2, you can have buttons specific to each template folder as well. This works very similar to the above custom images per template folder but also takes into consideration the language in use.

4.12.1 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. The original buttons are placed in "***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.

Note: To use different buttons for a specific template folder only, while leaving the default buttons for other sets, read the next chapter.

- 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)

4.12.2 Replace buttons for a specific template set

Since v4.4, each template set can use its own set of buttons. If a button is not found in the template set, the default osCommerce button will be used.

To replace a stock osCommerce button placed in catalog/images/, place a button image with the same name in:

catalog/includes/sts_templates/**mytemplates/images/english/buttons**/ and STS will use it (using “mytemplates” in this example as the template folder in use).

For example, if you are using the template set "multiple", English buttons are placed in catalog/includes/sts_templates/**multiple/images/english/buttons**/

For example, if you are using the template set "multiple", French buttons are placed in catalog/includes/sts_templates/**multiple/images/french/buttons**/

5 List of existing placeholders

5.1 General placeholders

Here are the tags that are supported by default in STS 4.6 ("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 **you don't need to use it in your template**. Note: 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.
$langid$	New in v4.5.8: This tag allows you to add the language ID to your image link to call an image according to the language selected by the user.

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 5.6 for details.

5.2 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
$tellafriendbox$	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.

5.3 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 in these areas of the code.

You can also use templates for the content part only. See example in chapter 4.8.2.

5.4 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. 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. NOTE: Do not put <title>...</title> tags in your header. It will be added as part of the $headcontent$ variable.
<!--$extracss-->	Optional. Allows placing a link to a stylesheet dynamically. Will be replaced by the content of $sts->template['extracss']. The whole html line must be placed in, like <link rel="stylesheet" type="text/css" href="stylesheet.css"> 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.
$warning_header$	Optional. Allows warning messages to be displayed. Created in includes/classes/sts.php.

5.5 Placeholders for footer

Tags generally put in the footer of the page, created in the file “catalog/includes/modules/sts_inc/**footer.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.

5.6 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 several new custom tags in STSv4.6.

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

Placeholder	Meaning
$ url_index$	URL link to the index.php page.
$ url_products_new$	URL link to the New Products page.
$ url_specials$	URL link to the Specials page.
$ url_contact_us$	URL link to the Contact Us page.
$url_advanced_search$	URL link to the Advanced Search page.
$url_reviews$	URL link to the Reviews page.
$url_conditions$	URL link to the Terms of Use page.
$url_privacy$	URL link to the Privacy page.
$url_shopping_cart$	URL link to the Shopping Cart page.
$catmenu$	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!

5.7 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 STS Product Info module in admin.

Placeholder	Meaning
$productid$	The ID of the product being displayed. Added in v4.0.6 for compatibility with STS v3. Buggy until v4.3. Removed in v4.3SP1, use $productsid instead.
$productsid$	The ID of the product being displayed. Added in v4.0.6 for compatibility with STS v3. Buggy until v4.3SP1
$startform$	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 striked).
$productname$	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.

<table>
<tr><td>$product_popup$</td><td>New in v4.0.7: The Javascript with small image and link to popup window. Empty if no product image.</td></tr>
<tr><td>$optionheader$</td><td>The "Available Options:" text above the options, will change with the language. Blank if no options (product attributes)</td></tr>
<tr><td>$optionnames$</td><td>A list of Option (Product Attribute) names, separated by line breaks ("<br>"). Blank if no options</td></tr>
<tr><td>$optionchoices$</td><td>A list of Option (Product Attributes) choices as drop-down boxes, separated by line breaks ("<br>"). Blank if no options</td></tr>
<tr><td colspan="2">NOTE: As shown in the “/catalog/includes/sts_templates/multiple/content/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.</td></tr>
<tr><td>$moreinfolabel$</td><td>The language-sensitive value of TEXT_MORE_INFORMATION (for linking to a more detailed page)</td></tr>
<tr><td>$moreinfourl$</td><td>The URL of the page where more information can be found</td></tr>
<tr><td>$productdatelabel$</td><td>Will contain either the Date Added or the Date Available messages, language specific</td></tr>
<tr><td>$productdate$</td><td>The date it will be available or the date it was added</td></tr>
<tr><td>$reviews$</td><td>The text with the number of reviews for the item. Empty if no reviews are available</td></tr>
<tr><td>$reviewsbutton$</td><td>The image for the reviews button</td></tr>
<tr><td>$reviewsurl$</td><td>The URL for the product's reviews</td></tr>
<tr><td>$addtocartbutton$</td><td>The button to send the product into the cart</td></tr>
<tr><td>$alsopurchased$</td><td>The box with products also purchased</td></tr>
</table>

5.8 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 were you want the system messages to be displayed.

5.9 Placeholders for popup_image.php template

The template for the popup image page has been introduced in v4.4 and will not work with older versions. Note that placeholders generated in general.php and sts_user_code.php are by default not available. If you want them, you have to add their names in the module configuration in admin.
The STS Popup Image module must be enabled in admin, otherwise the template will be ignored.

Placeholder	**Meaning**
$productname$	The product name. Can be use as a title on the template.
$productmodel$	The product model.
$popupimage$	The product image.
$back$	Previous image button. Reserved for cases where products have several images.
$next$	Next image button. Reserved for cases where products have several images.
$count$	To display something like 2/7 (image 2 on 7). Reserved for cases where products have several images.

6 Tips and tricks

6.1 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 ***'$templatedir$/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.

> **Note: Since STSv4.6**, you can use the **$templateidr$** (with the end stop character $) to include items such as images from your template folders

image folder. For example: Inserting this code into your template would pull in the image from your current template set:

img src="**$templatedir$**/images/checkout_bullet.gif"

It would be the same as using the following absolute URL:

img src="catalog/includes/sts_templates/*your template folder*/images/checkout_bullet.gif"

Using the **$templatedir$** variable allows you to make all of the URLs relative, therefore, giving you the power to create unlimited template sets that will work instantly, no matter what you have in your core osCommerce images folder. This is great for third party STS template designers to offer templates to the masses.

6.2 Save some execution time (Required as of v4.6)

6.2.1 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.

IMPORTANT!
Since STSv4.6, ALL variables should use a stop end character. Otherwise, some variables will not function correctly.

Modify your custom template pages to use a stop end character:

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. Each 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.

6.2.2 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.

6.2.3 GZip compression

Since v4.3.3 (4.3 SP3), GZip compression can be turned on in admin. Before this version, it would probably display a blank page.

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

Attention HTC users

> Do not use this hack if you are using the Header Tags Controller modifications to the catalog/index.php file. HTC already uses settings in the admin to dynamically add the page title and category descriptions.

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.

6.4 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. What could be easier?

6.4.1 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.

6.4.2 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.

6.5 Create your own placeholder file to include (optional)

6.5.1 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 if 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. If you put it in sts_user_code.php, all pages will run this code even when 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.

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. What could be easier?

6.5.2 How?

Put the code in a file and upload the file in the “includes/modules/**sts_inc**/” folder and configure the STS modules in the admin that will use this file.

6.6 Create Images per Language ID

6.6.1 How to have a different image per language in STS:

Say you want to have a different image used in your template dependent upon the language that the user has selected in your shop. This can be accomplished by doing the following:

A new placeholder (tag) was created in STSv4.5.8 and placed in the “includes/modules/sts_inc/**general.php** file. The new STS tag is called: **$langid$**.

Create the images you want dynamically chosen per language as so: myimage1.jpg, myimage2.jpg, myimage3.jpg, etc, depending on how many languages you have in your shop, and save them in your images folder (or any osC folder such as images/banners).

You can name the image anything you want but keep them the same, just change the number according to the ID of the language that it represents. In other words, you do not have to name them "myimage", you could name them "languagemenu1.gif" or languagemenu2.gif" and so on.

The number for the LangID will come from **your specific** osCommerce settings. To find the language ID for each of your shop's languages, go to your Admin, Localization, then select each language and view the ID in the URL.

Now add the following into your STS Template where you want the dynamic language image to appear: <img border="0" src="images/myimage$langid.jpg">

The STS tag "$langid$" will be replaced dynamically by the correct language id number that the user selects when viewing your shop.

> For example: english is ID=1 in my shop so the URL would be dynamically changed to "images/myimage1.jpg" if the user selected the language "english" in my shop.

7 Infobox templates

7.1 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**

Note: The infobox_header.php.html file is optional. This template allows you to have a custom infobox header, separate from the content header.

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
> **$content** - contents inside the infobox.

You can create any html-code for your infoboxes, simply by inserting in it the three labels above. **It is important to not use the end stop character "$" in any of the variables above in your infobox templates.**

Note: Since STSv4.6, you can use the **$templateidr$** (with the end stop character $) to include items such as images from your template folders image folder. For example: Inserting this code in your infobox template would pull in the image from your template set:

<img src="**$templatedir$/**images/checkout_bullet.gif">

It would be the same as using the following URL:

<img src="catalog/includes/sts_templates/*your template folder*/images/checkout_bullet.gif">

Using the **$templatedir$** variable allows you to make all of the URLs relative, therefore, giving you the power to create unlimited template sets that will work instantly, no matter what you have in your core osCommerce images folder. This is great for third party STS template designers to offer templates to the masses.

It is possible to create specific 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**

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

- infobox_***infoboxname***.php.html

If you want to edit the $content of the infobox, such as adding an image in front of each category listing in your infobox_categories.php.html template, you will need to edit the actual catalog/includes/boxes/categories.php file to have those changes reflected in your infobox template.

***Tip:** An easy way of modifying the very important category box would be to use one of the many category contributions available at osCommerce.com. Here are two that work very well with STS:

1. **Category Box as Nested Unordered** Listhttp://www.oscommerce.com/community/contributions,4201

2. **"Dynamenu" for osCommerce**
   http://www.oscommerce.com/community/contributions,4228

7.2 Box around product listing

The original product listing is using the boxes class to create the frame around the product listing. With that said, you would get undesirable results if you used the same infobox template like we do for the other boxes. To prevent this, up until STSv4.4, this part was not using the template system but the original osC design. Many users understandably felt the need to template this portion as well.

Since STSv4.4, a specific template can now be used for the product listing. It should be named **infobox_product_listing.php.html** and be placed in the same folder as each of the other infobox templates.

7.3 Boxes created by catalog script

Some catalog scripts like shopping_cart.php and advanced_search.php are also using the boxes class, so they'll use the default template and generate undesirable results. As boxes can have the same name as a catalog script (this is the case for shopping_cart.php), there must be a difference between templates for infoboxes or for a box in a catalog script. Since STSv4.3, the templates for catalog scripts are located at the same place as other box templates, **but are named differently**:

- **catalog_**scripname_header.php.html (optional)
- **catalog_**scripname.php.html

For example: The template set "multiple" includes templates for shopping_cart.php and advanced_search.php (**catalog_shopping_cart.php.html** and **catalog_advanced_search.php.html**).

Note: the shopping_cart.php page uses a box only when the shopping cart is empty. Try it.

Author of “Infobox Templates for STS”:
Grigory Kopanev (gkopanev@mail.ru), modified by Bill Kellum for use in STSv4.6.

7.4 Test Infobox Templates

Activate template 'multiple' in admin:
In Admin/Modules/STS/Default module, select “multiple” from the "Template Folder" drop down menu and activate infobox templates with "Use template for infoboxes" = true.

The infoboxes are now using the templates defined in /includes/sts_templates/**multiple/boxes**/.

If you want this feature in your custom “Template folder”, such as “my_templates”, “christmas” or “spring”, then all you would need to do is create a “**boxes**” folder in your custom template folder and then add your infobox templates into this folder. Of course, you will also need to enable the infobox template feature in the STS Default Module in the admin.

8 Support Forum

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

http://forums.oscommerce.com/topic/295660-contribution-sts-v4/page__view__findpost__p__1475994

Please post your STS related questions to this thread (use above link) so that the STS Community as a whole can benefit.
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:

site:forums.osCommerce.com "[Contribution] sts v4" "infobox"

The example above would search the STSv4 Contribution support thread for anything pertaining to “infobox”.

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. ☺

9 STS Power Pack

Be sure to visit the STS Power Pack site at http://www.oscommerce.com/community/contributions,4456 . This site is full of additional STS add-ons and helpful guides. If you are having trouble integrating a contribution with STS, then you may find your answer on this site. Several other users have posted their fixes and solutions to problems you may encounter.

Here is a short list of STS Add-ons available at the STS Power Pack site:

- Free STS Template Modern RC2a
- Header Tags Controller (STS Power Pack)
- Ez new fields (STS Power Pack)
- Master Products (STS Power Pack)
- STS Tips, Tricks and How To’s Volume One
- Dynamenu & STSv4
- Custom Header and Footer Includes for STSv4x (*STSv4.6 Multiple Template set uses this feature*)
- STSv4.3.3 and HTCv2.6.0 Bundle
- MoPics6 popup
- Add New Pages Using STS
- Top of page for STS
- QT Pro STS hack
- STSv4.5.2 and HTCv2.6.3 Bundle
- STS_MEGA_POWER_PACK_BUNDLE
- LIGHTBOX Image for STS 4+
- Attributes Sorter Copier for STSv4
- STSv4.5.2 and PRODUCT_EXTRA_FIELDS
- Dynamic Sitemap and STS
- STS and Marquee Text Banner
- Easy Meta Tags – CUSTOM Product Tags for STSv4
- And more….

> If you come up with some custom STS tags or solutions that you would like to share with the STS Community, then I urge you to post your STS Add-ons to the STS Power Pack site. I look forward to seeing your work!

10 About the Sample Templates

STSv4.6 comes out of the box with six (6) sample template sets (template folders) and one (1) blank template set. Each of these sample template sets should be considered as “starting points” for your own personal designs. They are provided “as is” and of course “Open Source”. They can be edited and modified anyway you see fit. The purpose of providing these sample template sets is to display “visually” what STS can do for your osCommerce designs.

Keep in mind that this is only a sampling of what STS can do. STS can add so many more features to your osCommerce store that depends upon your skill set in PHP and other web technologies.

10.1 Blank Template

The Blank template set is provided for the sole purpose of visually displaying the minimal code needed to make your custom templates function correctly in the Simple Template System. This template set should not be used on a live store until you create at least a default template named sts_template.html.

- One Blank Template (to be used as a starting point for your designs)

10.2 Blue Hash Template

Preview: http://www.soundsgoodrp.com/eshop

The Blue Hash template set is provided by Open Source Web Design (http://www.oswd.org). It has been converted to be used with STSv4.6 by Bill Kellum. The Blue Hash template set features the following:

- One Default Template
- One Default Infobox Template
- One Specific Infobox Template
- Custom Stylesheet
- Popup Image Template
- Custom Image Button Set
- Includes several custom page links
- Horizontal Menu Navigation

10.3 FreeosCommerce 009 Template

Preview: http://freeoscommerce.com/free-templates/view-template/?template=009

The freeosc_009 template set is provided by Zachary Coyne of freeosCommerce.com (http://www.freeoscommerce.com). The freeosCommerce 009 template set features the following:

- One Default Template
- One Default Infobox Template
- One Specific Infobox Template
- Custom Stylesheet
- Popup Image Template
- Custom Image Button Set
- Horizontal Menu Navigation
- Custom Footer Text in main template

10.4 FreeosCommerce 060 Template

Preveiw: http://freeoscommerce.com/free-templates/view-template/?template=060

The freeosc_060 template set is provided by Zachary Coyne of freeosCommerce.com (http://www.freeoscommerce.com). The freeosCommerce 060 template set features the following:

- One Default Template
- One Default Infobox Template
- One Specific Infobox Template
- Custom Stylesheet
- Available Custom Image Infobox Button Folder (if you wanted to design your own "corner_left.gif, corner_right.gif,etc." infobox images that would be separate of the main osCommerce infobox images).
- Custom Image Button Set
- Horizontal Menu Navigation
- Custom Footer Text in main template

10.5 FreeosCommerce 061 Template

Preview: http://freeoscommerce.com/free-templates/view-template/?template=061

The freeosc_061 template set is provided by Zachary Coyne of freeosCommerce.com (http://www.freeoscommerce.com). The freeosCommerce 061 template set features the following:

- One Default Template
- One Default Infobox Template
- One Specific Infobox Template

- One Default Product Content Template
- Custom Stylesheet
- Custom Image Button Set
- Horizontal Menu Navigation
- Custom Footer Text in main template
- Available Custom Image Infobox Button Folder (if you wanted to design your own "corner_left.gif, corner_right.gif,etc." infobox images that would be separate of the main osCommerce infobox images).

10.6 Multiple Template

The Multiple template set is the STSv4.6 version of the old "full" template set that was available in older STS releases. It has been modified by Bill Kellum (http://www.soundsgoodpro.com/eshop) to work with STSv4.6. The Multiple template set features the following:

- One Default Template
- One Default Category Template
- One Specific Category Template
- One Default Manufacturer Template
- One Specific Manufacturer Template
- One Specific Product Info Template
- One Popup Image Template
- One Default Infobox Template
- Two Specific Infobox Templates
- Two Specific Catalog Infobox Templates
- One Product Listing Infobox Template
- One Specific Product Content Template
- Custom Stylesheet
- One Custom Header Template
- One Custom Footer Template

10.7 Single Template

The Single template set is the STSv4.6 version of the old "test" template set that was available in older STS releases. It has been modified by Bill Kellum (http://www.soundsgoodpro.com/eshop) to work with STSv4.6. The Single template set features the following:

- One Default Template
- Custom Stylesheet

Note: This template set DOES NOT use the infobox template feature in STS. You will need to make sure this feature is not enabled in the STS Default Module in the admin.

STS v4.6 document by Bill Kellum (billkellum@soundsgoodpro.com) of Sounds Good Productions.