

eDatCat Customization Guide

Last Updated 2004-06-30: The eDatCat Knowledge Base is a developer's reference tool designed to help in the process of troubleshooting, understanding, and deploying the many tools that are built in to the eDatCat Shopping Cart System.

Introduction

This guide will walk you through the process of understanding the eDatCat templates, customizing them for your needs, and explaining some of the intricacies of dealing with a product that relies heavily on your creativity.

As you delve into the subject of the eDatCat templates, there are a few things you need to keep in mind. These principles will help you earn a great grasp on the templates and give you the ability to truly enjoy the flexibility and potential of this product.

- 1. Nothing is "canned".**
Sure the default templates are enough to power a basic store, but they are meant to be customized. This product is designed to allow you control over the presentation. EDatCat does not can the output of the shopping cart, the database, the templates, etc...it is all left to your creativity...designed to allow you to do your job as a developer. The key is to think creatively.
- 2. eDatCat is not designed as a quick-fix solution.**
Many developers look for a quick-fix solution for clients that need their store up and running within a few days. In most cases, eDatCat is not the solution for this situation. If you have used the product before and are familiar with it, then, yes, it can be done. We have customers who have designed entire sites around eDatCat- including a 60,000-item inventory- in just a few days. However, if this is your first time with eDatCat, plan for a couple of weeks in development. Because eDatCat is so flexible and relies on your creativity for the presentation, it takes a little time...all good things do.
- 3. Use the available resources.**
Our support site contains a wide variety of information- from instructional tips and manuals to articles, notes, discussions, and examples. Use these resources to help fuel your creativity and provide some guidance for your specific needs and ideas. If you get stuck, or if you cannot find an answer to your question on our support site, we are here to help. You can open a support ticket or contact us via email. Our support methods are proven to be very effective in addressing your needs.

OK, now you're ready to start learning the eDatCat templates and how to bend them to meet your specific needs. Let's get started!

How the eDatCat Templates Work

Date last updated: 5/4/2004

As we state over and over, eDatCat does not "can" anything for you, where possible. This product gives you control over output and presentation. In order to accomplish this measure of flexibility, there is a template for just about everything. You only need to use the templates you need...if you will not be using the wish list feature, you can disregard the two templates that control that aspect of the program (wishpage and wishrow). This is good since you are not required to customize every single aspect of the software just to use a few of its features.

eDatCat uses the templates for every screen the software generates. When you look at the templates...let's take the mainpage template, for example, you will notice that some things are hard-coded onto the template and other things are called with tags, or Shortcuts, as we call them. A shortcut is a tag offset by two percentage signs that calls information from the eDatCat database. On the mainpage template, you will see a shortcut called %%feature1%%. In this case, the %%feature1%% shortcut calls the html code for the first featured product and puts it in place of the tag when the screen is viewed online. The shortcuts are at the heart of eDatCat's flexibility. There are dozens of shortcuts available for use in countless ways. You can even create your own shortcuts if you want (we'll get to that later).

The pages generated by eDatCat are dynamic, meaning they are generated on the fly. It is important to note that you can use as many shortcuts as you want...or you can hard-code many things directly onto the template. You can use a single template with a variety of shortcuts to generate every product page, or you can use custom templates with hard-coded information for each product. You can also create your catalog on static html pages and simply reference the shopping process. There are a variety of ways to approach the presentation...let me suggest my preference.

I suggest that you use the shortcuts and the default templates wherever possible. Some information may need to be hard-coded, and that is fine. The advantage of using the shortcuts is that the information can be updated automatically by changes in the admin, rather than changing each individual page. This will make your life much easier...trust me.

Modifying these templates is simply a matter of making changes to the html code you find on them. Add your own graphic text, re-arrange the shortcuts, etc. The possibilities are endless. Whatever changes you make, eDatCat will use these templates to generate each screen in the shopping process.

These templates are designed to give you total control over the presentation of your web site. How good the site looks, how it is arranged, the color/graphic scheme, etc. ...that is completely in your hands.

Where to Start with the Templates

Date last updated: 6/10/2002

OK, so you have just installed eDatCat on your server. You're thinking, "great, what now?".

The following procedure will help streamline the process of configuring and customizing eDatCat to your needs.

- 1. Complete the essentials in the admin section.**

This will allow you to test your store with a small amount of test data. The essentials are:

- a. Complete the Program Configuration section of the Admin.
- b. Add a single category (Categories/Subcategories).
- c. Add a single product to the product database (Modify Product Info)
- d. Complete the Shopping Cart section of the admin.

- 2. Determine which eDatCat templates you will be using.**

One would assume that you have already laid out a basic site map to use as a guide in the development of your site. With this in hand, determine the templates you will need and which templates you will not need. **BE CAREFUL DOING THIS-** you don't want to get rid of a template that you need. **DO NOT** delete the un-needed templates...simply copy them over to another directory on your server for safekeeping until you are finished with your site. Once you have weeded out those templates you will not be using, make a backup of those you do need. It is helpful to have the default version of the template in case you run into a snag.

- 3. Walk through the order process with the default templates.**

This will give you a chance to see the software in action with the default setup of the templates.

- 4. Begin customizing the most basic templates.**

I have tried a number of combinations here, but the below steps make the most logical sense and will help you stay on track. In essence, you want to customize the templates in the order that they will be presented to the customer. Usually follow a pattern along the lines of the following:

- a. head-tags and body-tag
- b. mainpage, along with header and footer

Mainpage is the most important, generally speaking, because it outlines the basic layout and presentation for the rest of the site. I normally spend some time on this template and make sure it looks and feels the way I want it. I have navigational elements on each page, then I put the code for those links in a separate file and call it into the template with the file(path) shortcut. Regardless, by the time I am finished with mainpage, I have a page I can

as a guide for customizing the rest of my templates...thank god for copy & paste.

- c. category
- d. listpage and listrow
- e. detail
- f. cartpage, cartrow, empty-cart, and carterror
- g. order-login, login, and login-success
- h. acct-setup, acct-success, and acct-info
- i. checkout and order_rows
- j. billing and shipping

At this point, I am ready to work on my secure templates, payment and payerror. It is important to note that the of the templates in the checkout process (payment, payerror, and confirm_page), should be referenced from the secure URL...payment and payerror will actually reside on the secure server. Confirm_page will not, but will still be coded as if it were. This is also the time to go ahead and upload the xxxSSL.cgi script to the secure cgi-bin and set permissions to 755. You will also need to create a templates directory, upload the payment and payerror templates and set permissions on the files and directory to 777.

- k. payment and payerror
- l. confirm_page

5. **Walk through and test your customizations.**
You will have viewed and tested each template and you made customizations, but now it is time to go back and walk through the entire process to be sure all is in order. Place a few test orders just to be sure.
6. **Customize the rest of your templates.**
Now that you have the basics out of the way, you can go back and customize the rest of the templates you will be using.

This process should help you streamline your experience with eDatCat. If you have elements that will reside on each page, I cannot stress enough the importance of the file(path) shortcut. This shortcut is addressed later in this guide, but the intelligent use of it will help you immensely and save you all sorts of time as you go through and customize your store.

For more information, see:

- [User Manual notes on Where to Start](#)

How to Make the Most of the eDatCat Templates

Date last updated: 6/10/2002

Working with the eDatCat templates is really quite simple, once you have a good understanding of how it all fits together. It's a good idea to get your mind wrapped around the whole concept of the eDatCat templates, shortcuts, and functions before you dive too heavily into your project. This Customization Guide will help you do just that. Once you get the idea, it's a matter of keeping a few simple principles in mind:

- **Be thorough.**
With eDatCat, there are generally many ways to accomplish a single task. It helps to examine each of these ways and determine the way that best suits the needs of your project. This is key, since some ways are easier than others. With thorough planning, you can make things much easier on yourself and your client down the road by selecting a solid strategy for accomplishing the needs of the job.
- **Be careful.**
Because eDatCat is comprised of so many parts (it's really more of a toolbox....you build what you need), it's sometimes easy to get turned around or confused. Be careful of typos, system paths, IF...THEN statements

composition, etc. One tiny error in any of these areas and a problem could occur...sometimes, the solution is o right in front of you. Check the documentation in this Customization Guide if you get confused. Of course, if yc don't find the answer here, you can always contact Support.

- **Be creative.**

Often, we find that customers use a feature of the eDatCat templates in ways we never imagined possible. Thi comes from their creative approach...and we encourage this. eDatCat is highly flexible...it has features and use even we haven't discovered yet. So, gain a good understanding of the software, how it works, how it fits togeth and then create a template strategy that meets the needs of your job. A creative approach to the templates car open new possibilities and can help make your project more successful than you first conceived.

In short, you can accomplish the exact look & feel your project calls for with eDatCat. In the process, you will find it mu easier if you are thorough, careful, and creative.

Templates

A few words should be said about the header/footer structure of the templates. While there are the head-tags, body-tag, header, and footer templates available by default, you don't have to use them. The advantage of using them is that it makes updates to the site very easy- change one file and you're done. However, if you wish to make each page look different, you can do that. Or, if you have JavaScript on some pages that doesn't really work with the head-tags or body-tag used by the rest of the site, omit them on those pages and use them on the rest. It's very simple.

The head-tags, body-tag, header, and footer files are called with the file(path) shortcut. You will see this shortcut on the default templates. It is also used on the default templates to call the left-hand navigational bar.

You can use these files, or not. You could even create a variety of header and footer files and use some on some pages and others on other pages...it's totally up to you.

That said, let's dig into the templates and what they are used for.

Complete List of Templates

Date last updated: 6/10/2002

It would be helpful to you to print out the complete list of eDatCat templates. This list contains the file names and descriptions of each of the template files. This file is in PDF format.

[Download the list here.](#)

For more information, see:

- [Knowledge Base topics on Template Customization](#)

Creating Custom Templates

Date last updated: 6/10/2002

The default setup of eDatCat allows you to use a single category and detail template for each of your categories and products. However, if you have some categories or products that require a different or unique presentation, you can use a custom template for those categories or products. It's simple...

Copy the original template...for the sake of example, we'll use the detail template. Copy it and rename it to something else (leaving the original in the directory as well).

Next, customize that template to meet your needs.

Finally, in the admin, under the product record, provide the complete server path (not the URL) to your custom template where asked. Once saved, the script will now use your custom template instead of the default detail template. The same process works for categories, listpage, and others.

You can also use your own templates to present information outside the catalog itself. For instance, if you wanted to present your contact information, policy information, or even your home page on an eDatCat template, you can do so. In fact, you can power your entire site with eDatCat.

There are several advantages to this. Obviously, the eDatCat shortcuts and functions operate on eDatCat templates - not static pages. In addition, you can still use the IF...THEN Statements and File Shortcuts to expand the effectiveness of templates and make site updates very simple.

To create your own templates, the "link" function is used. In other words, create your template, with similar construction to other eDatCat templates, give it a name, and place it in the templates directory. Then, call your custom template with "user_action=link&link=yourtemplatename". That's easy.

Subdirectories can be used in your template directory for custom templates. All standard eDatCat templates MUST reside in the default template directory. For your own page template, you can place them in subdirectories of the default template location. You must only specify this in the link. For example, if you place a template called "contactinfo" in a subdirectory of the default template location called "mypages", you would call it using "user_action=link&link=mypages/contactinfo". That's that easy.

A good example of creative custom template use is our own web site, <http://www.edatcat.com>. Every page in our site is generated by eDatCat...even the home page. Take a look at our site for a few ideas.

For more information, see:

- [Making your mainpage template your home page](#)
- [Knowledge Base topics on Template Customization](#)

Incorporating Static HTML Pages

Date last updated: 6/10/2002

You can incorporate your static HTML pages into your eDatCat presentation in a variety of ways. Below is a brief summary of each method:

- **Use the link function to make your static pages eDatCat templates.**
See the article about Creating Custom Templates in this section of the Guide for details on how to do this. You rename your static page with the same extension as the rest of your templates if you go this route, but updates to your site will be much easier...plus, the shortcuts and functions are now available on your pages, whereas they

would not be if they were normal HTML pages.

- **Use the popup field to link to more information.**

There is more information on this method in the Special Instructions section of the Guide, but it basic allows you link to static pages or other templates in the product record. This is a good way to use static pages to enhance spice up your product detail templates.

- **Run your catalog entirely from static HTML pages.**

We don't suggest this, as it only creates more work for you, but if this is what you need to do, then you can, of course. You can reproduce all of the forms and information on the templates on a static HTML page and treat it like any other page on your site. If you run your catalog from static page though, be advised that the shortcuts functions will not work (they only work when called from within the script)....everything will need to be hardcoded. This method is generally advised only if you really know what you're doing.

Static pages have their positive points...know when, where, and how to use them in conjunction with the eDatCat temp and you will add value to your catalog.

For more information, see:

- [Creating Custom Templates](#)
- [Knowledge Base topics on Template Customization](#)

Important Note on Changing Admin Templates

Date last updated: 5/4/2004

In Release 3.0, all admin screens are now generated using templates found in the admin directory. These templates m modified to change the look of your admin interface.

When modifying these templates, great care should be taken, or you may introduce multiple problems in your shopping cart. In addition, **YOU MUST** follow the below outlined procedure **BEFORE** you make changes to the admin templates **YOU DO NOT** follow this procedure, and you experience problems as a result of your changes, then **WE WILL BILL HOURLY** to fix it. This policy is for your benefit as well as ours.

Before changing the admin templates:

1. Make a list of the changes you wish to make.
2. Submit this list to [eDatCat Support](#).
3. Support will review your list and make you aware of any issues to be cautious of. In addition, Support may advi the best way to approach your changes.
4. Once you receive this response from Support, you may proceed "with our blessing", and we will assist you if you encounter problems.

When you get ready to change the admin templates:

Make a backup of all of the default templates in your directory. Store them someplace safe.

Here are a few "do/don't" guidelines you will need to follow when modifying the admin templates:

Do Not's

1. Do not change or delete any shortcuts on the template.
example: %%backordered_list%%
2. Do not delete form tags.
3. Do not change or delete any hidden variables within the form tags.

Example:

```
<input type="hidden" name="user_action" value="prog_admin">
```

4. Do not change the names, values, or type of any form objects. Size and formatting may be changed.

Example:

```
<input name="test_mode_fl" type="checkbox" value="ON" %%test_mode_checked%%> (Do not change anything the form object.)
```

5. Do not change the name (including case) of template names.

Do's

1. Translate labels, headings, and instructions.
2. Change formatting, column layout, colors, and fonts.
3. Change button look.
4. Change look and size of input objects.
5. Add images, text, popups, and links.
6. Delete input fields and manually set them as hidden variables, only if absolutely positive of what you're doing.

Example: `<input type="hidden" name="template_style" value="full">`

This will ensure that template style never gets changed to the old style.

After making any changes, you should thoroughly test your cart.

For more information, see:

- [Knowledge Base topics on Template Customization](#)

Complete List of Admin Templates

Date last updated: 6/10/2002

It would be helpful to you to print out the complete list of eDatCat admin templates. This list contains the file names and descriptions of each of the template files. This file is in PDF format.

[Download the list here.](#)

For more information, see:

- [Knowledge Base topics on Template Customization](#)

Shortcuts

Within the templates, you may include any HTML, JavaScript, or other coding you desire. When you open any of the templates in a text editor, you will notice several custom tags and form functions. These elements are your "toolbox" and will help you create high-powered templates quickly and easily. Called shortcuts, these tools allow you to present any data you desire on your web pages. These shortcuts make it much easier to update your online content.

For example, you may have a featured product you want to display at your store entrance. Rather than opening your template and editing the html every time you change this featured item, you simply change it in your admin section by adding the catalog number of the item in the Featured Products section. On your mainpage template, you place the shortcut . This will call the featured item and display it on your template.

Another example is if you want to welcome your customers by name after they log in. All you have to is place html code similar to the following on your login-success template:

```
<p align="left"><strong>Welcome firstname(shortcut)!</strong></p>
<p align="left">We're glad that you returned to catname(shortcut). what would you like to do now?</p>
<ul>
<li>viewacct(shortcut)</li>
<li>viewwish(shortcut)</li>
<li>orderstat(shortcut)</li>
<li>shopping(shortcut)</li>
</ul>
```

If your business is named "Johnny's Apples", and the customer's name is Maggie, this code would produce the following message on the login-success template:

```
Welcome, Maggie!
We're glad that you returned to Johnny's Apples. What would you like to do now?
```

- View your account info
- View your wish list
- View your order status/order history
- Start Shopping!

The bulleted list, of course, would contain links to the various pages.

eDatCat uses the "Shortcuts" to quickly embed links and text within your templates. The Shortcuts CAN ONLY BE USED WITHIN YOUR EDATCAT TEMPLATES. Because these Shortcuts refer to routines and subroutines within the EDatCat program, they must be used on pages generated by the program...otherwise, they will not be recognized.

Some of these shortcuts are generated contextually. In other words, if you are using the Shortcut for the drop-down list of subcategories, this list will contain the subcategories specific to the category the user is browsing. When the user changes categories, the list of subcategories changes accordingly. The same goes for the featured products within a category, category title, additional product options, etc.

As you can see by the list (linked above), these shortcuts provide you with a huge amount of flexibility in creating high-powered templates while reducing the time you spend on creating them. In addition to these Shortcuts, EDatCat also provides you with a set of tools for building your own forms and links. These tools are called "Functions".

Complete List of Shortcuts

Date last updated: 6/10/2002

It would be helpful to you to print out the complete list of eDatCat templates. This list contains the file names and

descriptions of each of the template files. This file is in PDF format.

[Download the list here.](#)

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)

IF...THEN Statements

Date last updated: 5/4/2004

How many times have you said it? You know, "If only so-and-so, then thus-and-such." Countless times, I'm sure. The high-tech revolution of the past 4 years seems to have been partially spawned and was certainly carried by GenX Lollapalooza and Lillith Fair wannabe types that said "if only we could..." ...and then they did. Well, EDatCat isn't exactly the brain-child of this sort, but it does leave room for the starry-eyed Skywalker's of the webmaster world who are bright enough to envision the "then" beyond the "if" ("if they traced the robots here, they may have learned who they sold them to...then that would lead them back...home." "Wait Luke, it's too dangerous!"). Danger, schmanger...the EDatCat IF...THEN shortcut can be your greatest ally in creating a seamless, dynamic environment that constantly invents itself based on the criteria you supply.

The shortcut is designed to allow you to print something to your template IF certain criteria are met.

For example, IF the "popup" field has data present, THEN the shortcut should print the information on your template:

```
%%IF($popup ne "") THEN <small><em>To read what other customers have said about this product, <a href="$popup" target="_blank">click here</a>!</em></small>%%
```

This shortcut can be used in a variety of ways. Here are a few ideas:

- Print addtocart button if item is in-stock. If it isn't in stock, print a message explaining the situation to the customer.
- Print label or heading for the related items if related items have been assigned to the product,
- Print code for popup window if the popup field has data in it (above),
- Print a message stating that special handling fees apply if the price of a product is above a certain amount,

As you can see, the possibilities are endless. This shortcut takes up a lot of slack in managing the day-to-day affairs of e-business...especially when used in conjunction with inventory controls and price-sensitive fees and alerts.

Here are a couple of examples of this shortcut in action:

- **Add to Cart Button**

This is an example from our own demo site. This shortcut is used three times for a single object- the addtocart button. Appropriate actions are supplied for each of the "status" options being used on the site.

```
%%IF($status eq "In-Stock") THEN <input type="image" name="addtocart" value="submit" src="http://www.edatcatdemo.com/images/addtocart.gif" width="64" height="20" alt="addtocart" border="0">%%
```

```
%%IF($status eq "Out-of-Stock") THEN <font face="Verdana" color="#800000"><strong><small><small>We're sorry, but this item is currently out of stock.<br>Ordering is disabled now, but please check back soon.</small></small></font>%%
```

```
%%IF($status eq "Backordered") THEN <input type="image" name="addtocart" value="submit"
src="http://www.edatcatdemo.com/images/addtocart.gif" width="64" height="20" alt="addtocart"
border="0"><br><small><small><font face="verdana" color="#800000">This item is backordered, so there will
delay in delivery!</font></small></small>%%
```

- **Label for Related Items**

This shortcut states that if related items are present for this product, then print the appropriate label.

```
%%IF($related_items ne "") THEN Other Items to Consider:%%
```

- **More Info Link**

This application of the shortcut states that if there is value in the popup field, then a link to the page will be print

```
%%IF($popup ne "") THEN <small><small><font face="Verdana"><em>For additional details on this product, <
href="$popup">click here</a>!</em></font></small></small>%%
```

So how does the shortcut work?

The IF...THEN statement is comprised of two parts: the "expression" and the "result".

Examples of acceptable syntax:

```
%%IF(expr) THEN result%% or
%%IF(expr) THEN "result"%% or
%%IF(expr) THEN(result)%% or
%%IF(expr) THEN(result) ELSE(result)%%
```

The ground rules for the IF...THEN statements are:

- Enclose result strings in parenthesis when embedding shortcuts or for consistency.
- There can be no space between the IF or THEN and the following parentheses.
- There can be no carriage returns breaking the statement.
- There must be a space between components of the expression.
- Since parenthesis are part of the syntax, literal parenthesis must be enclosed in quotes, within parentheses.
- Can have up to 3 tests.

Operators

The first part of the statement, the expression, contains an operator, the element of the statement that makes some so comparison or validates some piece of information. See the below table for an explanation of the available operators.

Operator	Name	Description	Example	rest
Valid String Operators				
gt	string greater-than	Compares whether left value comes alphabetically after right value.	"fat" gt "hat"	fals
lt	string less-than	Compares whether left value comes alphabetically before right value.	"fat" lt "hat"	tru

ge	string greater-than or equal to	Compares whether left value is equal to or comes alphabetically after right value.	"pat" ge "pat"	tru
le	string less-than or equal to	Compares whether left value is equal to or comes alphabetically before right value.	"sat" le "pat"	fals
eq	string equality	Compares whether left value is exactly the same as the right value.	"bob" eq "bob1"	fals
ne	string inequality	Compares whether left value is at all different from the right value.	"bob" ne "bob1"	tru

Valid Numerical Operators (*Note there must be a space before and after symbol)

<	numeric less than		1 < 2.0	tru
<=	numeric less than or equal		1 <= 2	tru
>	numeric greater than		1 > 3.0	fals
>=	num. greater than or equal		1 >= 1	tru
==	num equal to (2 equal signs)		2.0 == 2	tru
!=	num not equal to		2.0 != 2	fals

between

Compound Test Operators

or	or	Any one expression must be true to be true	1 > 2 or "TAD" lt "ZOO"	tru
and	and	All expressions must be true to be true	1 > 2 and "TAD" lt "ZOO"	fals

Expressions

The Expression of the statement also contains the elements the operator is to work with. For instance:

example: %%IF(\$status eq "In-Stock").....

Expressions may contain:

- Variables
- String literals (must be enclosed in quotes)
- Numeric literals (cannot be enclosed in quotes)
- Product flags: \$new_item (YES,NO), \$discount_item (YES,NO)
- Account flags: \$whls_flag (YES,NO), \$logged_in ("1" if logged in)

Examples of expressions:

- !defined(\$firstname) (\$firstname has no value)
- \$qty_on_hand > 0
- \$status eq "Back Ordered"
- (\$qty_on_hand > 0 or \$status eq "On Order")
- \$qty_on_hand between 1 and 5

Results

The result segment of the IF...THEN contains information as to what will happen based on the criteria in the expression

example:THEN()%%

Results may include:

- Variables
- Shortcuts
- File shortcut
- Text and HTML
- Space separating dollar sign and price field \$ \$price

Examples of result values:

- (Welcome, %%firstname%%)
- Your price only \$ \$price!
- (%%file(%%templatedir%%sizechart.txt)%%)
- ("(This Item takes 48 hours to ship)")

Complete Examples:

(No carriage return breaking statement.)

1. %%IF(defined(\$lastname)) THEN(If you're not %%firstname%% %%lastname%%, Click here!)%%
2. %%IF(\$qty_on_hand between 1 and 3) THEN Supplies are limited!!)%%
3. %%IF(\$logged_in eq "1") THEN(%%file(%%templatedir%%acct-welcome.txt)%%) ELSE(Please login to our site. Or register now!)%%
4. %%IF(\$new_item eq "YES") THEN()%%

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)
- [Knowledge Base topics on IF...THEN Statements](#)

File Shortcuts

Date last updated: 6/10/2002

One of the beauties of working in a Unix environment for me has always been the flexible use of SSI, or server side includes. I use them for a variety of things, not the least of which involves storing headers, footers, and other static code in exterior text files. This means that I can update various aspects of my pages by editing a single file rather than deal with each page individually.

With the first version of eDatCat, the templates are extremely customizable, but the age-old problem of editing volumes files comes into play when you have navigational elements or other things that need to be placed within the body content of each page. Now, we have made this a snap with a new shortcut that allows you to simulate SSI flexibility.

NOTE: In the examples below, the short format has been altered...dashes have been placed around the percent signs of the shortcuts (to keep them from rendering in this manual). The actual shortcuts do NOT have dashes around the percent signs.

The shortcut is:

```
%%file(server-path-to-file)%%
```

Let's say that I have a navbar in the left margin of each of my pages. Rather than hard coding the navigational elements onto each EDatCat template, I can use this new shortcut and cut my development and update time to a large degree.

The format of this shortcut is very simple. If I am using this for my navbar, then I would place the html code for the navbar in a text file and name it "navbar". I would place this file in my eDatCat templates directory, giving it the same file extension as my other templates.

Next, on each of my templates, I place the file shortcut:

```
%%file(/usr/home/domain/www/templates/navbar)%%
```

or, if my template has a .txt extension...

```
%%file(/usr/home/domain/www/templates/navbar.txt)%%
```

This will call the contents of my navbar file onto the page where the shortcut is placed. That's all there is to it!

This shortcut can be used in hundreds of ways. It's a simple concept, but takes the flexibility and customization potential of the EDatCat templates just one step further.

Enjoy!

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)

Creating Custom Shortcuts

Date last updated: 5/4/2004

In the data directory, there is a text file called XXXtmpl.edc. This is where most of those cool shortcuts are stored. You can edit this text file and add your own shortcut! The format is simple-- shortcut name followed by the shortcut text, separated by a double colon. You can even have a shortcut within a shortcut. Remember that one type of shortcut is any data field enclosed in %%. So your new shortcut could include your customer's first name.

Example:

`product_push::%%firstname%%, this %%item%% was made for you!`

Just remember that shortcut names are like variable names -- spaces or crazy characters do not work.

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)

Collecting Custom User Information

Date last updated: 3/22/2004

Sometimes you will want to collect custom information about your customers, in addition to the usual name, address, etc. In general, you can do this using the five CUSTOM shortcuts. They are: CUST1, CUST2, CUST3, CUST4, CUST5.

You use these shortcuts the same way as you would any other, i.e. you do not need to know more than the usual HTML. Just start making use of these variables.

Collecting custom information:

Instead of using NAME="state" to collect information (in this case, state), you would use NAME="CUST1", etc.

e.g.

`Tax Exempt ID: <INPUT NAME="CUST1" SIZE="20">`

Displaying custom information:

On pages where you display customer information, you would use %%CUST1%%, etc.

e.g.

`Your tax exempt ID is %%CUST1%%`

Using IF...THEN statements with custom information:

On pages where you use IF...THEN statements, you can refer to the variable as \$CUST1, etc.

e.g.

`%%IF($CUST1 eq "") THEN(You have not entered your Tax Exempt ID)%%`

Example 1: Where did you hear about us?

When an account is being set up, in addition to name, address, etc. you wish to ask your customer, "Where did you hear about us?"

One common way to set this up is to provide a drop-down menu, with a text box for specifics. For example,

```
<SELECT NAME="CUST1" SIZE=1><OPTION SELECTED VALUE=" "please specify...
<OPTION VALUE="Google">Google
<OPTION VALUE="Sales Associate">Sales Associate
<OPTION VALUE="Word of mouth">Word of mouth
<OPTION VALUE="Other">Other (please specify)
</SELECT>
<p>
If other, please specify: <input NAME="CUST2" SIZE="19">
```

In this case, CUST1 stores whether the customer found the site via Google, by being contacted by a sales associate, fi word of mouth, etc. If your customer selects "Other" and fills in the text field, CUST2 will contain the contents of that text field.

Templates where user information, such as address, is requested include:

```
acct-info
acct-setup
billing
shipping
```

Category Tree-Specific Shortcuts

Date last updated: 4/4/2004

As of eDatCat Version 3.0.5e, new shortcuts are available within the Category Tree Shortcuts (%%cat_tree%%). These include:

%%lct_category%% category of the category being listed

%%lct_template%% template of the category being listed

%%lct_feature%% feature of the category being listed

%%lct_logo%% logo of the category being listed

subcatsrow Template

Date last updated: 6/30/2004

In addition to listing subcategories in the category template, eDatCat Version 3.0.5e can now be enabled to use a subcatsrow template. This template also has access to subcategory-specific fields. In order to maintain backwards compatibility, this feature must be enabled by default.

(1) Download the [subcatsrow template](#). Save it with your template extension, i.e., txt, htm, html, etc. in your templates directory. Confirm permissions of the file.

(2) To enable this functionality, add the following line to your ABCsetup.edc file:

```
ENABLE_SUBCATSROW_TEMPLATE::1
```

Note that the variable must be in all uppercase and there are two colons between the variable name and the number 1

(3) Modify your subcatsrow template as you wish. The following extra shortcuts are available in this template.

%%sct_subcategory%% - category shortcut specific to the subcategory

%%sct_template%% - template shortcut specific to the subcategory

%%sct_feature%% - feature shortcut specific to the subcategory

%%sct_logo%% - logo shortcut specific to the subcategory

%%sct_ct_discount%% - ct_discount shortcut specific to the subcategory

%%sct_catdescr%% - descr shortcut specific to the subcategory

%%sct_catvtrate%% - catvtrate shortcut specific to the subcategory

%%sct_show_products%% - show_products shortcut specific to the subcategory

Functions

Like the Shortcuts, eDatCat "Functions" can be used within your templates to create a custom and convenient shopping environment. The benefit to you, the developer, is that the Shortcuts and Functions reduce the time it takes you to create such an environment.

However, unlike the Shortcuts, which can only be used within the eDatCat templates, Functions can be used anywhere—whether it be a dynamically generated page or a static page on another web site.

Understanding Functions

eDatCat interprets the Functions as commands. In other words, you link to the program and tell it what to do. You tell eDatCat that it is about to receive a function by providing it with "user_action=".

NOTE: In the examples below, the short format has been altered...dashes have been placed around the percent signs of the shortcuts (to keep them from rendering in this manual). The actual shortcuts do NOT have dashes around the percent signs.

A function can be used in a hyperlink as follows:

```
%%-script_loc%%-%%?user_action=addtocart&catalogno=%%-catalogno%%-
```

Notice that "addtocart" is a function. In this link, you are telling eDatCat to add an item to the shopping cart. This link would be used on an eDatCat dynamically-generated page and the Shortcut %%-catalogno%%- would be the catalog number of the item you wish to add.

Another example of a function used in a hyperlink is:

```
http://www.yourdomain.com/cgi-bin/mystore.pl?user_action=addtocart&catalogno=ITEM01
```

This link could be on another web site linking back to your store. This link would be on a static page and is providing EDatCat with all of the information it needs to add your item to the shopping cart.

Functions can also be used as hidden fields in a form. For example:

```
<form method="post" action="%-%script_loc%-%">  
<input type="hidden" name="user_action" value="login">
```

Again, this form uses Shortcuts, so it would be placed within an eDatCat template. If this form were on a static page, you would need to hard-code the path to the script in the form action.

In this instance, the Function "login" is used to tell eDatCat that the information submitted in this form should be used to log the user in. Of course, in this form, the text fields for the user's email address and password would be provided.

Functions allow you to create a strong measure of convenience for your customers. By using Shortcuts and Functions throughout the templates and the rest of your site, a truly interactive and easy shopping process is accomplished.

Complete List of Functions

Date last updated: 6/10/2002

It would be helpful to you to print out the complete list of eDatCat functions. This list contains the file names and descriptions of each of the functions. This file is in PDF format.

[Download the list here.](#)

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)

Function Reference

Date last updated: 6/10/2002

We have made available a reference guide to functions in PDF format. This material expands on the complete list and into more detail about using the eDatCat Functions.

[Download the list here.](#)

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)

Using the "Link Function"

Date last updated: 6/10/2002

The link function allows you to incorporate other pages into your eDatCat catalog presentation. You can even power your entire web site, home included, with eDatCat because of this function.

The syntax is simple:

```
user_action=link&link=yourtemplatename
```

Where "yourtemplatename", you would of course place the filename of your own template. There are a couple of ground rules, however. They are as follows:

1. The template must have the same extension as your other eDatCat templates.
2. You must provide the full path to the file in the link. If your file is in the default template directory, then you're all set. However, if you placed the file in another directory, then you must give the path in the link, relative to the default template directory.

example:

```
user_action=link&link/mypages/yourtemplatename  
user_action=link&link=/mypages/myfiles/yourtemplatename
```

3. You can use any available shortcuts and functions on these pages.

That's it. For a good example of this in action...our own web site is completely powered by eDatCat.

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)
- [Making your mainpage template your home page](#)

Special Instructions

JavaScript, CSS, and Flash

Date last updated: 6/10/2002

JavaScript

JavaScript can be used in countless ways. The principle here is that you can use JavaScript on eDatCat templates in the same manner as you would use it on a standard HTML page. Here's an example.

Let's say you want to use an image rollover script in your navigation menu. This is a two-part script, with the script itself residing in the head of the html, and the rest of the code buried throughout the body of the page.

In the header.txt template, place the script:

```

<script language="JavaScript">
<!--

if (document.images) {
img1on = new Image();           // MouseOver Images
img1on.src = "http://www.domain.com/images/iamgeon1.gif";
img2on = new Image();
img2on.src = "http://www.domain.com/images/imageon2.gif";

img1off = new Image();         // MouseOut Images
img1off.src = "http://www.domain.com/images/imageoff1.gif";
img2off = new Image();
img2off.src = "http://www.domain.com/images/imageoff2.gif";
}

function imgOn(imgName) {
if (document.images) {
document[imgName].src = eval(imgName + ".on.src"); }}

function imgOff(imgName) {
if (document.images) {
document[imgName].src = eval(imgName + ".off.src"); }}

//-->
</script>

```

Then, in your leftnav template, assuming that is where your navigation menu will be, place the other javascript code along with your images:

```

<a href="http://www.domain.com/cgi-bin/xxxstore.cgi?user_action=link&link=yourtemplate"
onMouseOver="imgOn('img1')" onMouseOut="imgOff('img1')">
</a>

<br>

<a href="http://www.domain.com/cgi-bin/xxxstore.cgi?user_action=link&link=yourtemplate"
onMouseOver="imgOn('img2')" onMouseOut="imgOff('img2')">
</a>

```

That's it! If your javascript is used on some pages, but not on others, or if you have various javascripts that are used on a variety of pages in mixed & matched manner, then you can employ the use of multiple header.txt templates.....simply create appropriate header for each page. It's a snap!

Cascading Style Sheets (CSS)

CSS is a wonderful tool. It is used in the default eDatCat templates to control link colors on the templates. However, it can be used to specify everything from page fonts to page background colors/images to table or form properties....it is a powerful tool in the development of a web site. There are three ways to use CSS.

- In-Line: this means that you basically place style properties within a specific tag, with no need of a "stylesheet"

Example:

```
<div align="left">
```

```

<table border="0" cellpadding="0" cellspacing="0" width="700" height="42"
style="border-right: 1px solid rgb(0,0,0)">
<tr>
<td height="42">this produces a right-side black border 1 pixel wide</td>
</tr>
</table>
</div>

```

This produces a right-side black border 1 pixel wide.

- Standard Style Sheet: this places the style sheet, which can contain a variety of style parameters, in the head of the html page.

Example:

```

<head>
<title>Standard Style Sheet</title>
<style type="text/css">
  a:link
  {
    color: #000000;
    text-decoration: none;
  }

  a:active
  {
    color: #000000;
    text-decoration: none;
  }

  a:visited
  {
    color: #000000;
    text-decoration: none;
  }

  a:hover
  {
    color: #000000;
    text-decoration: underline;
  }
</style>
</head>

```

This style sheet controls links on the page...the code above will produce black links that become underlined when the mouse moves over them.

- External Style Sheet: this is a great way to make page updates very simple. Place the style sheet shown above in a text file, and name it "style.css". From there, just include the reference to the style sheet in the head of your template or html page as shown below:

Example:

```

<head>
<title>External Style Sheet</title>
<link rel="stylesheet" type="text/css" href="http://www.domain.com/templates/styles.css">
</head>

```

This is the method employed by the default eDatCat templates. Using this method, you can easily update the properties for your entire site by changing the styles.css file. Just as with javascript, you can use multiple styles (standard or external) on each page or by using different header files.

Flash

Flash is a great tool when used properly. Since it is called as an embedded object, it can be used very easily on your eDatCat templates. Some of our customers are working with using the eDatCat shortcuts and functions within flash to create a very interactive experience...more information on the specifics of this will come in the future.

For a great demonstration of an eDatCat web site using flash, see [Salsa Express](#). This site shows with great skill how can be used in combination with eDatCat to present an impressive shopping environment. This site also uses a great of DHTML, JavaScript, and CSS.

Multiple Columns on the ListPage Template

Date last updated: 5/4/2004

eDatCat can be configured to use multiple columns on the listpage/listrow template. You can use as many rows as you tricky, but quite simple once you get the idea.....here goes...

Under Sorting/Listing in the eDatCat admin, you will find a setting for the number of columns in the product listing. For this example, we'll assume that you are using 2 columns.

Normally, in the listrow template, you would have html code for a complete table, such as:

```
<div align="left"><table border="0" cellpadding="0" cellspacing="0" width="700">
<tr>
<td width="350"><font face="Arial" color="#FF0000">%%catalogno%%; %%item%%</font></td>
<td width="350"><font face="Arial" color="#FF0000">%%viewmore%% or %%addtocart%%</font></td>
</tr>
</table>
</div>
```

This would produce:

```
%%catalogno%%; %%item%%                                %%viewmore%% or %%addtocart%%
```

This table would be looped for each item in the product list.

Now, in the event you use more than one column, eDatCat steps in a helps with the code. Since the columns are created in addition of <tr> tags, the script prints them for you...it will print these tags to produce the number of columns that you si

This means that you have to place a little extra code on your listpage template, to coincide with the code that the script prints. Again, let's assume that you want to use 2 columns for your product listing.

This said, on your listpage template, you need the following:

```
<div align="left"><table border="0" cellpadding="0" cellspacing="0" width="700">
%%item_rows%%
</table>
</div>
```

Obviously, this code is not complete.

But wait....on your listrow template, you need the following:

```

</td>

<div align="left"><table border="0" cellpadding="0" cellspacing="0" width="700">
<tr>
<td width="350"><font face="Arial" color="#FF0000">%%catalogno%%; %%item%%</font></td>
<td width="350"><font face="Arial" color="#FF0000">%%viewmore%% or %%addtocart%%</font></td>
</tr>
</table>
</div>

</td>

```

Notice that the code in blue is your regular listrow html code, as shown above. However, the code is still not complete. Remember that eDatCat will print the <tr> tags to produce the number of columns that you specify. These final tags, which will complete the html code, are printed by eDatCat when it renders the page in the browser. This will produce the following, assuming a 2 column layout:

```

<tr>
<td>
<div align="left"><table border="0" cellpadding="0" cellspacing="0" width="700">
<tr>
<td width="350"><font face="Arial" color="#FF0000">%%catalogno%%; %%item%%</font></td>
<td width="350"><font face="Arial" color="#FF0000">%%viewmore%% or %%addtocart%%</font></td>
</tr>
</table>
</div>
</td>
</tr>
<tr>
<td>
<div align="left"><table border="0" cellpadding="0" cellspacing="0" width="700">
<tr>
<td width="350"><font face="Arial" color="#FF0000">%%catalogno%%; %%item%%</font></td>
<td width="350"><font face="Arial" color="#FF0000">%%viewmore%% or %%addtocart%%</font></td>
</tr>
</table>
</div>
</td>
</tr>

```

The <tr> tags printed by the script automatically are shown in blue. This will produce a 2-column produce list.

It's confusing, I know.....but, believe it or not, it's the best way to give you total control over the number of columns per your product listing.

The good news is that you can use this same method to control the number of columns for the search results, the most listings, the what's new listings, and even the related items!

It's a little hairy, but study it, test it, work with it.....you'll get it.

Popup Windows

Date last updated: 5/4/2004

The popup field in your eDatCat database (rendered on the templates using %%popup%%) is basically just a place to |

link to more information about a product. If you want an actual popup window, you have to place the code yourself. H we can't do everything for you....that would defeat the purpose we've always held of giving you total control. But it's a p of cake.....check it out.....

Let's say you have a product and want to link to a page that shows a few customer comments. You must create this comment page yourself...name it whatever, place it wherever...it doesn't matter. Then, place the complete URL to this in the popup field in the product record (that's Modify Product Info for your newbies).

On your detail template (or listrow), you will want to place this link using the %%popup%% shortcut. Since the shortcut only writes the URL, you will need to do the rest. eg:

```
<a href="%%popup%%">Click here for customer comments</a>
```

To just open the page in a new window, then simply add:

```
<a href="%%popup%%" target="_blank">Click here for customer comments</a>
```

Not really a popup window, but will work for some of you.

For those of you that want a real-life popup window, try this:

In the head of your document, or in the header.txt template, place this script:

```
<script LANGUAGE="JavaScript">
<!-- Begin
function popUp(URL) {
day = new Date();
id = day.getTime();
eval("page" + id + " = window.open(URL, "" + id + "",
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=0,width=400,height=200,left = 300,top =
200');");
}
// End -->
</script>
```

In the body of your document, or on the detail or listrow template, place this link:

```
<a HREF="javascript:popUp('%%popup%%')">
See what other customers have said about %%item%%
</a>
```

Now, what if you only have a popup window for certain products, but not for others? That would produce alot of dirty a unneeded code on your template. No worries.....the IF...THEN statement comes to the rescue...

```
%%IF($popup ne "") THEN <a HREF="javascript:popUp('%%popup%%')">
See what other customers have said about %%item%%
</a>%%
```

Remember that IF...THEN statements have to be on a line all by themselves. If the statement is broken, or if it wraps t another line, it will not work. For more info on the IF...THEN statements, see that section of this guide.

Hopefully this has given you a few ideas on using the popup field in the eDatCat database. The possibilities are endless but, hey....that's the point, right?

Search Forms

Date last updated: 5/4/2004

EDatCat allows you to build your own search forms to suit your needs. You can build your own form in the following fo

```
<form action="%%script_loc%%" method="post">
<input type="hidden" name="user_action" value="search">
<input type="hidden" name="screen" value="1">
<textarea rows="2" name="descr" cols="20"></textarea>
<input type="submit" value="search">
</form>
```

You can include whatever fields in this form you wish from the product file. You can also optionally specify a sort order for the results by including an additional hidden field such as:

```
<input type="hidden" name="sortorder" value="catalogno">
```

Keep in mind, though, that searches on the catalogno field are not really searches- they will return an exact match...this how most searches for a catalogno works on other sites as well.....so, you can't combine the catalogno with other field the same form.

We normally recommend have a search form for catalogno that allows the customer to enter a specific and exact catalogno. Another form can be used to search the description and item fields, etc.

We have included detailed instructions within the search templates themselves as well....see the commented portions in the HTML code of these templates for more detailed instructions.

Using Account Flags

Date last updated: 5/4/2004

From time to time, you may find yourself in need of presenting certain links or information to customers who are logged in. To do this, eDatCat has to determine whether or not they are logged in. No problem.

By using certain "account flags", you can tell eDatcat to check to see if a user is logged in and, if they are logged in, to present a link or welcome message. Also, you may want to give wholesale customers one link and retail customers a different link. Let's take it a step at a time. We'll give you a few examples and then let you take it from there.

Checking a customer's login status...

To check to see if a customer is logged in, you must check to see if there is a value in one of their account fields. Obviously, if they are not logged in, then the script does not know who they are....if they are logged in, then it does. So, let's say you want to print a welcome message to customers who are logged in.

To do this, you need to first check to see if they are logged in, and then, dependant upon the answer, print a message. This requires an IF...THEN statement such as:

```
%%!IF($logged_in eq "1") THEN>Hello %%firstname%%>%%
```

Very simple. But what if they are not logged in? By using a second statement, you can take this one step

further...

```
%%IF($logged_in eq "1") THEN(Hello %%firstname%%)%%
%%IF($logged_in eq "0") THEN(Hello Please <a href="%%login%%">Please login</a>)%%
```

"Nifty", huh? You can also, in addition to the logged_in flag, check to see if there is value in any of the other customer account fields such as firstname, userid, lastname, etc.

Checking a customer's wholesale/retail status...

To print a message to wholesale customers, the script needs to know if they are indeed a wholesale customers. Once again, an IF...THEN statement is needed. For example:

```
%%IF($whsl_flag eq "YES") THEN(<a href="whatever">Click here to see the latest wholesaler news</a>)%
%
```

Great...but what if they are not a wholesale customer? Yeah, yeah....add another statement...

```
%%IF($whsl_flag eq "YES") THEN(<a href="whatever">Click here to see the latest wholesaler news</a>)%
%
```

```
%%IF($whsl_flag eq "NO") THEN(<a href="whatever">Get a life</a>)%%
```

You can use this to print links, messages, news, or whatever else. For more info on the IF...THEN statements, see the section of this guide.

Multiple Columns on the ListPage Template (Method 2)

Date last updated: 4/4/2004

Currently, eDatCat supports using multiple columns on the listpage template by modifying the templates, as described [Multiple Columns on the ListPage Template](#) section of the [Customization Guide](#). In addition to this, as of Version 3.0.5e (Release Date: April 5, 2004), eDatCat can accommodate multiple columns without any changes to the default templates

Because this functionality would break backwards compatibility for customers who may have modified their templates, not enabled by default.

To enable, in your ABCsetup.edc file, add the following line:

```
ENABLE_CODE_MULT_COLUMNS::1
```

Note that you have to use uppercase and there must be two colons between the variable name and the number 1.

Product Option Style Parameters

Date last updated: 4/4/2004

Currently, the most convenient way to change the styles of drop-down and text product options is to modify the perl scripts. As of Version 3.0.5e, you can add extra parameters without modifying the scripts.

(1) You have to enable the use of optional parameters. For drop-down product options, you can do this by adding the following line to your ABCsetup.edc file:

```
ENABLE_DISPOPTION_PARAMS::1
```

For text product options, you do this by adding the following line to your ABCsetup.edc file:

```
ENABLE_DISPTOPTION_PARAMS::1
```

Note that you must use UPPERCASE for the above variables and there must be two colons between the variable and the number 1.

(2) To specify your drop-down product options, you may use the following variables in your ABCsetup.edc file:

- `dispooption_params::Your drop-down product option parameters`

e.g. To have your dropdown product options in italics with purple text on a blue background:

```
dispooption_params::style="color:#FF00FF;background-color:#0000ff;font-style:italic;"
```

- `disptoption_params::Your text object parameters`

e.g. To have your text product options in regular font with red text on a blue background:

```
disptoption_params::style="color:#FF0000;background-color:#0000ff;"
```

Note that the `dispooption_params` and `disptoptions_params` are lowercase and that you need to use two colons between variable names and the parameter.

SSL and Payment Gateways

Secure Server Templates

Date last updated: 6/10/2002

Whenever you are using a secure server to serve up pages, you have to be careful to reference everything securely so avoid those pesky "This page contains unsecure information" alerts. Since the eDatCat-driven pages are built from several components, or templates, a few words of guidance may be needed...

Let's assume that your payment template utilizes the header, footer, head-tags, leftnav, and body-tag templates. In order for the payment template to be secure, all of these other templates must be secure as well. The solution is simple: copy them all over to the secure server and make the needed adjustments.

What are the needed adjustments? Simple....here are a few things to look for:

- On the payment template, be sure that the file shortcuts that call these other templates are referencing a secure directory.
- On all of these templates, check image tags, style sheet references, and any other references to embedded objects to be sure that all of them are pointing to a secure URL. For instance:

Instead of: ``

You need: ``

And instead of: `<link rel="stylesheet" type="text/css" href="http://www.domain.com/templates/styles.css">`

You need: `<link rel="stylesheet" type="text/css" href="https://www.domain.com/templates/styles.css">`

- Also, and I'm sure this goes without saying, be sure that all of the images, style sheets, and other objects you are referencing securely actually reside on the secure server.

Typically, your secure server directory structure may look something like this:

`/templates/`

`payment (or au_payment or whatever payment template you need)`
`payerror`
`header`
`footer`
`head-tags`
`body-tag`
`leftnav`
`styles.css`

`/images/`

`whatever images the above named files reference`

`/cgi-bin/`

`xxxSSL.cgi`
`xxxlib.cgi`
`xxxsetup.edc`

`/data/`

`xxxerrmsg.edc`
`xxxstoremsg.edc`
`xxxtempl.edc`

For more information on configuring your secure server, including file permissions, admin settings, etc., see the User's Manual, SSL Setup section.

Regular SSL, No Payment Gateway

Date last updated: 5/4/2004

(1) Under **Checkout Configuration**, go to **Payment Options**. Under **Secure Server Options**, select **Local SSL Site** or **Remote SSL Site**, as appropriate. Under **Payment Processing Service**, select **None**.

(2) Under **Card Types Accepted**, select which credit cards you wish to offer, adding any you wish to offer that are not already listed.

(3) If you wish to collect the V-check digit, i.e., the three or four digit number on the back of the credit card, check the **Enable V-check** checkbox.

(4) Under **Checkout Configuration**, go to **SSL Setup**. You will be prompted with the SSL configuration screen for either **Remote SSL** or **Local SSL** as appropriate.

- **Remote SSL:**

(5) Under **Secure Server Location**:

URL to SSL script: This is the secure server (https:) url of the SSL script.

e.g. `https://secure.server.com/mydomain/cgi-bin/ABCSSL.pl`

Path to SSL cgi scripts: This is the complete path to the scripts directory on the remote SSL server.

e.g. `/usr/home/mydomain/www/cgi-bin/`

Path to Template Directory: This is the complete path to the templates directory on the remote SSL server.

e.g. `/usr/home/mydomain/www/templates/`

(6) Under **Payment Notification Options**, select how you wish for payment information to be stored. If you select yes to "Save Orders?", credit card information will be stored in the `ABCorders.edc` file in your data directory.

Path to Data Directory: This is the path to your data directory. If you are using Remote SSL, it will be the path to the data directory on the SSL server.

Path to Payment Upload Directory: This is the path to your uploads directory.

URL to Payment Upload Directory: This is the URL to your payment uploads directory.

(7) Under **SSL Email Setup**:

SSL Email Program: Choose the appropriate email configuration for your SSL server.

Mail Dir: Complete path to the email application on the secure server (if applicable).

e.g. `/usr/sbin/sendmail`

Mail Server: Email server on the SSL server.

e.g. `smtp1.mydomain.com`

(8) Copy your **SSL**, **chkout** and **lib** scripts and your **setup.edc** file to the secure server.

- **Local SSL:**

(5) Under **Secure Server Location**:

URL to SSL script: This is the secure server (https:) url of the SSL script.

e.g. <https://www.mystore.com/cgi-bin/ABCSSL.cgi>.

Path to Template Directory: This can be an alternative location on your server for templates.

(6) Under **Payment Notification Options**, select how you wish for payment information to be stored. If you select yes to "Save Orders?", credit card information will be stored in the ABCorders.edc file in your data directory.

Path to Data Directory: This is the path to your data directory.

Path to Payment Upload Directory: This is the path to your payment uploads directory. If you do not specify a path, the uploads directory will be used.

URL to Payment Upload Directory: This is the URL to your payment uploads directory. If you do not specify a url, the uploads url will be used.

SSL Email Payment Info: Choose between Regular Email, PGP (**NOT** supported), and None.

Regular SSL, With Payment Gateway

Date last updated: 5/4/2004

Whether or not you can or should use your own SSL server with your payment gateway depends on which payment gateway you are using and how it works. See the payment gateway section for details of your specific payment gateway. If you are not using a payment gateway documented here, please contact [EDatCat Support](#) and we will see if your payment gateway can be made to work with EDatCat.

Below are instructions for SSL configuration, regardless of your Payment Gateway setup. Note that you do not need SSL configured for some payment gateways and some of the following may be ignored, depending on your payment gateway.

(1) Under **Secure Server Options**, select **Local SSL Site** or **Remote SSL Site**, as appropriate. Under **Payment Processing Service**, select **None**.

(2) Under **Card Types Accepted**, select which credit cards you wish to offer, adding any you wish to offer that are not already listed.

(3) If you wish to collect the V-check digit, i.e., the three or four digit number on the back of the credit card, check the **Enable V-check** checkbox.

(4) Under **Checkout Configuration**, go to **SSL Setup**. You will be prompted with the SSL configuration screen for either **Remote SSL** or **Local SSL** as appropriate.

- **Remote SSL:**

(5) Under **Secure Server Location**:

URL to SSL script: This is the secure server (https:) url of the SSL script.

e.g. `https://secure.server.com/mydomain/cgi-bin/ABCSSL.pl`

Path to SSL cgi scripts: This is the complete path to the scripts directory on the remote SSL server.

e.g. `/usr/home/mydomain/www/cgi-bin/`

Path to Template Directory: This is the complete path to the templates directory on the remote SSL server.

e.g. `/usr/home/mydomain/www/templates/`

(6) Copy your **SSL**, **chkout** and **lib** scripts and your **setup.edc** file to the secure server.

- **Local SSL:**

(5) Under **Secure Server Location:**

URL to SSL script: This is the secure server (https:) url of the SSL script.

e.g. `https://www.mystore.com/cgi-bin/ABCSSL.cgi`.

Path to Template Directory: This can be an alternative location on your server for templates.

For more information on **Payment Notification Options** and not using any payment gateway, refer to the [Regular SSL Payment Gateway](#) section.

AuthorizeNet

Date last updated: 4/3/2004

(1) Under **Checkout Configuration**, go to **Payment Options**. Under **Payment Processing Service**, select **SIM (AuthorizeNet)**.

(2) Under **Checkout Configuration**, go to **Payment Gateway Setup**. You should now be prompted for the SIM Authorize.Net parameters.

AuthorizeNet Login: Your login ID as supplied by Authorize.Net

AuthorizeNet Transaction Key: Your Transaction Key as provided in the "Obtain Transaction Key" option under Sec in Settings.

Transaction Type: AUTH_CAPTURE to authorize and capture the funds for the transaction. AUTH_ONLY to authorize not capture the funds.

Description for Invoice: Any description of your choice for the invoice.

Company: The name of your company.

Email Customer Receipt? Whether or not you wish for Authorize.Net to email a receipt to the customer.

Email Merchant Receipt? Whether or not you wish for Authorize.Net to email a receipt to you, the merchant.

Merchant Email Address: E-mail address of the merchant.

Email Receipt Header: Text you would like to see in the header of your receipt.

Email Receipt Footer: text you would like to see in the footer of your receipt.

Test Mode: TRUE for test mode and FALSE for non-test (live) mode.

(3) In **Authorize.Net** Settings (on the Authorize.Net webpage):

Under **Response/Receipt URLs**,

Default Receipt URL: Leave blank

Default Relay Response URL: <http://www.yourstore.com/cgi-bin/ABCchkout.pl>

Under **Silent Post URL**, use:

<http://www.yourstore.com/cgi-bin/ABCchkout.pl>

Under **Relay Response URL**, use:

<http://www.yourstore.com/cgi-bin/ABCchkout.pl>

(**Relay Response URL** should already be set from the previous setting in **Response/Receipt URLs**.)

Under **Silent Response**, do not define any.

WorldPay

Date last updated: 6/8/2004

(1) Under **Checkout Configuration**, go to **Payment Options**. Under **Secure Server Options**, select **No SSL site will utilized**. Under **Payment Processing Service**, select **Other--I will supply parameters manually in the payment template**.

(2) Under **Checkout Configuration**, go to **Payment Gateway Setup**. You should now be prompted Other Gateway Settings.

(3) Fill in the following parameters:

Payment Accepted:

Field Name: transStatus

Success Value: Y

Error Message:

Error Message Field Name: RawAuthMessage

Session ID Field Name: M_sessionid

Optional Return Fields:

For other fields, you can enter any optional fields you want to pass through World Pay to include on your merchant or customer email confirmation, e.g. shipping instructions. Just name the field beginning with M_ and enter it in the other 1 text boxes. Include the value as a hidden field or form input object on the checkout page. Display it on your email as % %.....%%

EDatCat checkout template:

(4) Insert the following into the checkout template, being sure to replace **%%payaction%%** and **%%payfields%%** shor

```
<form action="https://select.worldpay.com/wcc/purchase" method=POST>
<input type=hidden name="instld" value="your installation ID here">
<input type=hidden name="cartld" value="%%sessionid%%">
<input type=hidden name="M_sessionid" value="%%sessionid%%">
```

```

<input type=hidden name="cost" value="%%grand_total%%">
<input type=hidden name="currency" value="USD">
<input type=hidden name="desc" value="Shopping Cart Items">
<input type=hidden name="testMode" value="100">
<input type=hidden name="name" value="%%firstname%% %%midname%% %%lastname%%">
<input type=hidden name="address" value="%%address1%%&#10;%%address2%%&#10;%%city%% %%state%%">
<input type=hidden name="postcode" value="%%zip%%">
<input type=hidden name="country" value="%%country%%">
<input type=hidden name="tel" value="%%hphone%%">
<input type=hidden name="email" value="%%userid%%">
<input type=hidden name="lang" value="EN">
<input type=hidden name=hideCurrency>

```

Notes:

- (i) Be sure to use your correct installation ID, currency code and language code. These are indicated in red above. The above html bit is already configured to use US dollars and English. Other currency and language codes can be found in Appendices of the WorldPay [Select Junior Integration Guide](#).
- (ii) Later, when you go live, you will need to delete the **testMode** hidden variable. Alternatively, you may set its value to (zero).

World Pay Customer Management System:

(4) Go to the **Installations** section. Select **Configuration options**.

(5) Enter the description fields as you would like them to appear.

(6) **Merchant's shop URL:** Enter your store URL here, of the form: www.mystore.com/cgi-bin/ABCstore.pl

Note that instead of ABC in the url above, you must use your store's three letter dbcode.

(7) For **Callback URL**, enter the URL to the checkout script, e.g. <http://www.mystore.com/cgi-bin/ABCcheckout.pl>

Note that instead of ABC in the url above, you must use your store's three letter dbcode.

(8) Check **Callback enabled** and **Use callback response**.

(9) Do not enter anything in the password fields. If there is something there, check the "use default" box to reset it. At the time, eDatCat is not programmed to check password fields.

(10) Click **SAVE CHANGES**.

(11) Using **Payment Page Editor**, you can customize the look of your WorldPay payment pages and the WorldPay bar that appears on your confirmation page.

(12) Test.

Cybercash

Date last updated: 4/4/2004

If you wish to configure Cybercash, please contact [eDatCat Support](#).

Echo

Date last updated: 4/4/2004

If you wish to configure ECHO, please contact [eDatCat Support](#).

VeriSign

Date last updated: 4/20/2004

PayFlowLink:

(1) Under **Checkout Configuration**, go to **Payment Options**. Under **Payment Processing Service**, select **VeriSign-PayFlowLink**.

(2) Under **Checkout Configuration**, go to **Payment Gateway Setup**. You should now be prompted for the Verisign Li parameters.

Under **Verisign PayFlowLink Parameters**:

VeriSign Login: This is your Verisign Login ID.
Partner: Use partner assigned by VeriSign
Email Customer: TRUE
Email Merchant: TRUE
Tx Description: A description of your choice.

Test Mode: TRUE or FALSE depending on whether or not you're testing.

(3) In VeriSign Manager, set the following:

Return URL Method: POST
Return URL: <http://www.yourdomain.com/cgi-bin/xxxstore.cgi>
Silect POST URL (checked): <http://www.yourdomain.com/cgi-bin/xxxchkout.cgi>
Transaction Process Mode: LIVE or TEST, depending on whether you are taking orders or testing.

For Return URL, you may use any URL you would like, except do not use the chkout script. You may, if you wish, go to different area of the store, use a static page, etc.

Be sure Transaction Process Mode in Verisign corresponds to whether or not you are using Test Mode in EDatCat.

Although the confirmation email should be sent out, you cannot display a confirmation page including order details, etc

Note: If you get a "The METHOD field is empty and is required" error, please see this [KnowledgeBase entry](#).

PayFlowPro:

If you wish to configure Verisign PayFlow Pro, please contact [eDatCat Support](#)

PayPal

Date last updated: 4/3/2004

(1) **Payment Options** in the EDatCat admin utility:

Under **Secure Server Options**, select **No SSL site will be utilized**; under **Payment Processing Service**, select **Non**

(2) In the **chkout** template:

Remove the `%%payaction%%` and `%%payfields%%` shortcuts and surrounding form tags. Replace with the HTML belc

```
<small><small>PRESS PAYPAL BUTTON TO PAY WITH YOUR CREDIT CARD USING THE
SECURE SERVICES OF PAYPAL. YOUR ORDER IS NOT
ENTERED UNTIL YOU HAVE COMPLETED YOUR PAYMENT AT
PAYPAL.
```

```
<form method="post" name="paypal" action="https://secure.paypal.com/cgi-bin/webscr">
<input type="hidden" name="Payprocess" value="OTHER">
<input type="hidden" name="amount" value="%%grand_total%%">
<input type="hidden" name="item_name" value="Shopping Cart Purchases">
<input type="hidden" name="item_number" value="%%invno%%">
<input type="hidden" name="business" value="email address of your paypal account">
<input type="hidden" name="cmd" value="_xclick">
<input type="hidden" name="return" value="%%scripturl%%dbcode%%chkout%%cgiext%%?sessionid=%%invno%
<input type="hidden" name="cancel_return" value="%%scripturl%%dbcode%%store.cgi?user_action=link&link=ca
<input type="submit" name="submit" value="Pay with PayPal">
</form>
```

Notes:

- The business parameter must be set to your paypal email address.
- Original message and item_name parameter can be adjusted as you prefer.
- We are using Paypal's aggregate method, whereby the aggregate order with the final total price is posted to pa
- Paypal users will have to return to your store after making payment for the order to go through on EDatCat's sic
- Sendemail can be set to NO if you don't want the customer to receive email confirmation.
- Paytype is optional, it is just descriptive and can be placed on the merchant email.
- You won't be using the payment template with this Paypal configuration.

(3) Create a template for the cancel page and call it **cancel**.

Authorize.Net and Paypal

Date last updated: 6/30/2004

Authorize.Net Configuration:

(1) Configure your SSL according to the instructions in the [Regular SSL, With Payment Gateway](#) section.

(2) Configure Authorize.Net according to the instructions in the [Authorize.Net](#) section.

Paypal Configuration:

This section is modified to take into account the Authorize.Net configuration.

(3) In the **chkout** template:

Keep intact the `%%payaction%%` and `%%payfields%%` shortcuts and surrounding form tags, for the Authorize.Net section. Add the HTML below:

```
<small><small>PRESS PAYPAL BUTTON TO PAY WITH YOUR CREDIT CARD USING THE
SECURE SERVICES OF PAYPAL. YOUR ORDER IS NOT
ENTERED UNTIL YOU HAVE COMPLETED YOUR PAYMENT AT
PAYPAL.
<form method="post" name="paypal" action="https://secure.paypal.com/cgi-bin/webscr">
<input type="hidden" name="Payprocess" value="OTHER">
<input type="hidden" name="amount" value="%%grand_total%%">
<input type="hidden" name="item_name" value="Shopping Cart Purchases">
<input type="hidden" name="item_number" value="%%invno%%">
<input type="hidden" name="business" value="email address of your paypal account">
<input type="hidden" name="cmd" value="_xclick">
<input type="hidden" name="return" value="%%scripturl%%dbcode%%chkout%%cgiext%%?sessionid=%%invno%
<input type="hidden" name="cancel_return" value="%%scripturl%%dbcode%%store.cgi?user_action=link&link=ca
<input type="submit" name="submit" value="Pay with PayPal">
</form>
```

Notes:

- The above html is different from that of the [Paypal-only](#) configuration.
- The business parameter must be set to your paypal email address.
- Original message and item_name parameter can be adjusted as you prefer.
- We are using Paypal's aggregate method, whereby the aggregate order with the final total price is posted to paypal.
- Paypal users will have to return to your store after making payment for the order to go through on EDatCat's site.
- Sendemail can be set to NO if you don't want the customer to receive email confirmation.
- Paytype is optional, it is just descriptive and can be placed on the merchant email.

(4) Create a template for the cancel page and call it **cancel**.

Other Gateways

Date last updated: 6/30/2004

EDatCat can be made to work with other gateways. If you have any questions or would like any help, do not hesitate to contact [EDatCat Support](#) and we will attempt to help you configure a new payment gateway.

NOTE: While we will help you configure an unsupported payment gateway, we cannot offer the same level of support as we can to more supported gateways.

HTML Editors

Using HTML Editors with eDatCat

Date last updated: 6/10/2002

Macromedia Dreamweaver

Date last updated: 6/10/2002

For more information, see:

- [Free eDatCat Dreamweaver Extension](#)

Microsoft FrontPage

Date last updated: 5/4/2004

I've had several inquiries as to whether or not FrontPage can be used with EDatCat. The answer, in short, is yes. However, given the ticky-tacky nature of FrontPage, a couple of adjustments need to be made.

We are all aware of the advantages of using a WYSIWYG (pronounced "wizzywig" and stands for "what-you-see-is-what-you-get") editor to design layout for web sites. In recent years, WYSIWYG editors have made even the most modest of creative minds a web guru. A WYSIWYG editor allows you to place objects and text on your page visually- much like Microsoft Word, Publisher, and other familiar desktop applications- without dealing with HTML code. However, the problem with WYSIWYGs is the fact that the software has to do your code-thinking for you. The software must be able to determine what it thinks you meant to do rather than what you actually do. This is generally no great hassle... in fact, it is quite helpful for many novices. However, the all-knowing WYSIWYG tends to trip and stumble a bit when it is presented with advanced code or tags. Ah yes, therein lies the rub.

The flexibility of EDatCat is built around a system of shortcuts and functions. The shortcuts, which are simple tags such as `%%item%%`, are placed within the HTML code of your EDatCat templates. These tags, or shortcuts, call data from your EDatCat database. For instance, if you are working on your product detail template, you place the shortcut `%%item%` where you want the product name to appear. All in all, EDatCat has some 60+ shortcuts and even lets you design your own. It is the use of these shortcuts that gives you such great control over the presentation of your online catalog.

When you use FrontPage to design and customize your EDatCat templates, the WYSIWYG nature of FrontPage just doesn't take kindly to these shortcuts when they are contained within variables of certain tags. For instance, if you place the `%%picture%%` shortcut directly on the page, FrontPage will look the other way... no questions asked. However, if you should be so bold as to place the `%%picture%%` shortcut in an image reference tag (e.g.- `), then FrontPage will assume that you meant to place the HTML code for the percent sign instead of the percent sign itself. So, instead of the example above, you'll get:

```

```

Man, what a headache. Everytime you save your template, change editing modes, or take a break, FrontPage will cha this code. So, how do you work around that?

In order to outsmart the WYSIWYG, you have to save your template in a more relaxed application, such as the ever-dr but oh-so-useful NotePad. Since most of you are probably saving your templates with a .txt extension instead of a .htm extension anyway, this is very simple. In the FrontPage Explorer, rather than double-clicking the template to open it, ri click it and choose "Open With" and "NotePad". This will open your template in NotePad. Copy your code over the to HTML view of the FrontPage Editor, and edit away! You can change editing modes all you want. FrontPage will still cl those shortcuts buried within HTML tags, but don't panic. When you get ready to save your template, go to the HTML v of the FrontPage Editor, copy your code (only the part you need for your template), and paste it back into NotePad. Th scan through that dirty HTML and remove those FrontPage SNAFU's, and then save your template in NotePad.

Wah-lah... all there is to it!

Adobe GoLive

Date last updated: 6/10/2002

NetObjects Fusion

Date last updated: 6/10/2002

eDatCat would like Orville Ludwig, of [Netexpressway](#), for submitting this article.

About the hardest part of integrating EdatCat into NOFMX is selecting your site style. A few simple steps will allow you edit and publish directly from NOFMX without having to recheck your html in an external editor after setup has been completed.

First time users follow the instructions for downloading and installing EdatCat on your server, when you get to the secti naming your template directory name it html (NOF likes this file name). After installation check to make sure everything working OK. If all is OK then using your favorite FTP program download the entire site to your local computer (back it u You will be using some of these files during setup when you begin building your site.

Open NetObjects and create a new site, while in site view add a new page and name it mainpage and name the page %-%catname%-% (EDatCat shortcut). For now mark the home page as a "do not publish" page. We don't want to overwrite any existing index pages during the build and testing process. Next go to publish view click on publish, arran files and select "by Asset Type"; this will put all of your pages in the html directory created during initial setup. If you ar already an EdatCat user and have named your template files something else, no problem. Still in publish view create a file folder and name it to whatever name you used when you first went through the setup process. Then move all the h files into the new folder before publishing.

While in publish view create a new folder and give it a name of your choice; this is where you will copy some of the files backed up. Close NOF and open your file manager, locate your backup directory and copy the following files to the ne

created folder in NOF's user sites your site:

- Catrow, confirmrow, econfirm, footer, header, head-tags, listrow, mconfirm, order_rows, ordstatrow, searchrow, wishrow and ABClib.cgi. These files will need to be edited in an external editor. (Notepad works well for most of them).
- Open footer, header and head-tags in notepad, delete everything and save it as an empty file. Next open ABClib.cgi and find the following section:

```

sub write_html_page      {

# combines headtags,body, passed strings, and footers
my($string,$cookie)=@_;
$htmlcode=<HTML><head>;
HTML:      {
    local $/;
    local *HTML;
    my($hfile)=$tplatedir."head-tags$tmplex";
    open (HTML,"< $hfile\0") || &CgiDie("00168","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>.</head>;
    close HTML;
    $hfile=$tplatedir."body-tag$tmplex";
    open (HTML,"< $hfile\0") || &CgiDie("00168","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>;
    close HTML;
    $hfile=$tplatedir."header$tmplex";
    open (HTML,$hfile) || &CgiDie("00169","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>;
    close HTML;
    $htmlcode.=$string;
    $hfile=$tplatedir."footer$tmplex";
    open (HTML,$hfile) || &CgiDie("00170","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>.</body></html>;
    close HTML;
}
$htmlcode=&shortcut($htmlcode);
$htmlcode=~s{ } { defined({$1}) ? ${1} : "" }gxex;
# $htmlcode=~s{ } { ${1} }gxex;
print header() if !defined($cookie);
print header(-cookie=>$cookie) if defined($cookie);
print "$htmlcode</html>";
exit(0);
}

```

And modify it to:

```

sub write_html_page      {
# combines headtags,body, passed strings, and footers
my($string,@cookies)=@_;
$htmlcode="";
HTML:      {
    local $/;
    local *HTML;
    my($hfile)=$tplatedir."head-tags$tmplex";
    open (HTML,"< $hfile\0") || &CgiDie("00168","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>."";
    close HTML;
    $hfile=$tplatedir."body-tag$tmplex";
    open (HTML,"< $hfile\0") || &CgiDie("00168","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>;
    close HTML;
}

```

```

    $hfile=$templatedir."header$tmplex";
    open (HTML,"$hfile") || &CgiDie("00169","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>;
    close HTML;
    $htmlcode.= $string;
    $hfile=$templatedir."footer$tmplex";
    open (HTML,"$hfile") || &CgiDie("00170","Writing HTML","Could not open $hfile","Yes");
    $htmlcode.=<HTML>."";
    close HTML;
}
$htmlcode=&shortcut($htmlcode);
$htmlcode=~s{ } { defined({$1}) ? {$1} : "" }gxex;
#$htmlcode=~s{ } { {$1} }gxex;
#print header() if !defined($cookie);
print header(-cookie=>[@cookies]);
print "$htmlcode";
exit(0);
}

```

There are only four lines that are modified.

Almost done, now open NOF and go to assets view and add all the files in your new folder as assets and check the pul always box. Start building your site by adding pages in site view; you will have to name all pages exactly the same as are named in your backup copy. Don't forget to use the eDatCat shortcut %-%catname%-% for the page title. Refer to backups for general site layout.

If you are not happy with the way some of the list rows are displayed you will have to edit them in an external editor to get them the way you want. The following is a list of the rows:

Catrow, confirmrow, econfirm, listrow, mconfirm, order_rows, ordstatrow, searchrow, wishrow

May sound a little strange but I like to use FrontPage to edit these files.

Although not tested this procedure should work the same in NOF 5

Misc Tips

Way-Cool Shortcuts

Date last updated: 6/29/2004

Why don't they just call it "the way"?

-Ellison Belt and Barbara Pettway

Mar 4, 2001- A shortcut is, by nature supposed to present a slight challenge. Remember Chevy Chase in Vacation? ...driving across the desert in the Griswald family truckster with a road map plastered to the steering wheel? Right, enough said. Think about it...if the shortcut were easier, it would not be called a shortcut...it would just be "the way"...that is, unless you are dealing with EDatCat shortcuts. We have removed the challenge and have made the shortcut the most convenient way to go.

The whole point of the EDatCat shortcuts are to give you control over the presentation of your online catalog. Our man! has always been "don't 'can' anything....leave it to the designer." As a designer, you need the ability to present various of data in a variety of ways from page to page. Our shortcuts give you this ability.

Most of the EDatCat shortcuts are obvious and self-explanatory. They give you an enormous amount of control over yc catalog design. Did we stop there? Of course not. We went a step further.

Barbara Pettway, our resident Perl guru and the programming mind behind the genius of EDatCat, has provided some down-to-earth answers to a few "infrequently asked questions" about the EDatCat shortcuts below...

Q. eDatCat shortcuts are soooo cool! How can I make my own?

A. Easy, Bud. In the data directory, is a text file called XXXtmpl.edc. This is where most of those cool shortcuts are sto! You can edit this text file and add your own shortcut! The format is simple-- shortcut name followed by the shortcut text separated by a double colon. You can even have a shortcut within a shortcut. Remember that one type of shortcut is a data field enclosed in %%. So your new shortcut could include your customer's first name.

Example:

`product_push::%%firstname%%, this %%item%% was made for you!`

Just remember that shortcut names are like variable names--spaces or crazy characters do not work.

Q. The above example would look stupid if the customer hasn't logged in yet! How can I correct that?

A. Very good point. You could use the IF...THEN shortcut either as part of your shortcut or on the template itself.

`Example: %%IF ($firstname ne "") THEN $product_push%%`

Q. Since my site is in the U.S. I don't need Euro prices, can I use this function to display Canadian dollars inste

A. Yes you can. See our demo store for an example. Just provide the multiplier in your admin menu. If you know the Canadian and US price for a product, get the multiplier by dividing the US price by the Canadian price. In other words, US price is 14.95 and the Canadian price is 20.95, then the multiplier would be: .713603819.

Q. I don't want to use your state and country table.

A. You can create your own state and country table and label them whatever you want on the templates, just don't cha! the file name or layout. The files are XXXtax.edc and XXXcountry.edc

Q. I don't want to use a state table at all, I want them to enter it in. Or, I don't want to use a country table, all my business is domestic.

A. Okay, Okay. Just get rid of it then. Replace the shortcuts with the appropriate HTML tags to collect the fields. The fie names should be state and country. (On the shipping address template they are called shstate and shcountry) Remem make the changes on all four templates acct-info, acct-setup, billing, and shipping.

Q. How can I display the retail price in strike-over if there is a discount?

A. Sure...see example:

`%%IF($discount ne "") THEN(Price: Regular price: <s>%%ret_price%%</s>
 Discounted Price: %% discount_price%%)%%
`

This would result in:

Regular Price: ~~\$50.00~~

Discounted Price: \$35.00

Q. That was cool! Can I do that if I have wholesale prices for distributors?

A. If they are logged in, use the whsl_flag field like this:

```
%%IF($whsl_flag eq "YES") THEN(Retail Price: <s>%%$ret_price%%</s><br> Your price: %%whsl_price%%)%%%
```

This would result in:

Retail Price: ~~\$50.00~~
Your Price: \$35.00

So, you see, the eDatCat shortcuts really are "the way". Odds are, if you want to do it, it can be done with eDatCat.

For more information, see:

- [Knowledge Base topics on Shortcuts and Functions](#)

How quick and easy is eDatCat?

Date last updated: 6/10/2002

The tool is only as good as the hand that wields it.

- Ellison Belt

I started developing web sites back in 1995, when the Internet boom was beginning to take America by storm. Amidst hype, pomp, and circumstance of the early days, the idea that the Web makes everything faster, easier, and cheaper emerged as a theme among hardware and software companies worldwide. The idea that a business can setup a web and instantly earn millions in sales revenue was a myth that deceived many a business owner into jumping at the opportunity to set up a quick and simple web presence. However, that "field of dreams" mentality that "if you build it, it will come", was soon diminished by the reality of business. Any successful business plan requires work and patience and solid business practices. Even Wall Street, after a few years of pumping millions into half-witted ideas scratched on a napkin, is finally coming full circle to embrace this reality.

In the past 5 years, I have learned a few things that have made my business very successful...

- It's hard to package a quality web site and sell it for pennies. Every client is different. Ultimately, you lose money when you package things too much. Besides, clients appreciate knowing they are not being viewed through a cookie-cutter lense.
- It's hard to survive in a competitive industry while doing things the quick & easy way. Excellence is earned and requires hard work, diligence, creativity, and patience.
- Software powers the behind the scenes functions....you have to be the master of the visual presentation and interface...any other method short-changes the project and puts limits on the creativity you bring to the job.

When it comes to building lower-end web sites (by this, I imply a budget of less than \$20,000), the same principles hold true. Even though the cost of developing a smaller e-commerce web site is less than, say, the multi-million dollar system touted by Amazon.com, the same principles of work, patience, research, methodology, and quality must be applied in proportion. Consider these facts:

- The current browse-to-buy ratio consistently hovers around 4%. This means that 4 out of 100 people that visit your site may purchase something.
- Over 85% of all shopping carts are abandoned before your customer makes a payment.
- When a visitor hits your home page, you have exactly 4 seconds to make a positive impression or they will leave.
- At last count, consumer satisfaction with online customer service was 18%.

All of these facts imply a few things. Keep in mind that the below conclusions are my own opinions. However, I happen to believe that they are correct. :) ...

- Customers are not able to quickly find what they are looking for.
- Customers use shopping carts for a variety of reasons- including to compare price totals for a number of products from different stores.
- Customers do not have time or wish to make time to decipher a complicated web site....they want to find something and they want it now.
- Customers expect to be treated well whether they purchase at your physical location or at your web site.
- Business owners are not taking e-commerce seriously enough.
- Business owners often look for a cheap, easy entrance to e-commerce...often from a "friend of a friend." This results in a poorly done web site with software solutions that do not cut the mustard. You get what you pay for. Even the internet can't overcome this basic fact of life.

These implications do not signal the end of the Web as we know it. In fact, e-commerce revenues continue to grow at a rather astounding pace. In addition, more and more business are going online and setting up e-commerce web sites. The difference today, as opposed to the situation a couple of years ago, is that businesses are now being forced to take this whole "e-thing" seriously. If they want to succeed and do the job right, they have learned to take care of business on the front end.

In the past two years, I have had a number of clients who called me in frustration and have told horror stories of paying thousands of dollars for a web site that a two-year old could have built. Others tell of a situation in which their friend's daughter threw together a few web pages at Geocities or even at home using something like MS Word or Publisher...obviously not up to par with even the lowest professional design techniques. They call me because they now have to start over and do things the right way...and they're willing to spend the bucks to do it. This is good news because it signifies a more down-to-earth approach to e-commerce.

Many of the problems facing today's web consumer are the responsibility of business owners and webmasters. Too many sites are thrown together in a peace-mill fashion. On top of that, many do not have the tools or the software resources to generate a quality web site on a reasonable small business budget. That, oh patient one, is why we created EDatCat to give you, the webmaster, control over the environment...and the business owner control over the services and products offered to the customer.

So where does EDatCat fit into all of this?

When I began putting together the behavior of the software in my mind, I realized very quickly that it would be a departure from the standard approach to cgi-based shopping carts. I was then faced with a choice...

1. Do I want a shopping cart that can be setup from a-z in under 30 minutes?
2. Or do I want a shopping cart that requires a little more time but that allows me the power to meet my clients' diverse needs?

I'll take door number two for \$350, Alex. It's a no-brainer. Don't forget those statistics outlined above. Much of the blame for this has been placed on consumer wariness about e-business. I disagree with this assessment. I believe that if the business offered a quality presentation that gave quality service, then the consumer would be more apt to trust him or her.

EDatCat does not promise that you'll start taking orders in 30 minutes. It doesn't even promise to deliver a quality presentation. In fact, EDatCat does not claim to provide you with a presentation at all. All it promises is that you will have the ability to do your job as a webmaster...to create a seamless environment rich with your client's identity and products...simple enough that even the silliest clown amidst the dumb-masses can navigate it. EDatCat gives you a touch of functionality. What you do with it and how you use it is totally up to you.

Can you create an entire web site in 30 minutes? Depends on how fast you work. I know of several customers who I set up and configured an entire 30,000 item store in under 3 days. Not too shabby. You see, my friend, EDatCat is different than other cgi scripts you are accustomed to. It does not can the output and allow you to work around it. It does not limit you to a certain design set or template layout. Sure, it comes with a default set of templates, but these are designed to be totally changed and altered to fit the design of your site...they are meant as a guide. We do not intend for you to use our templates "as is". EDatCat takes care of the behind the scenes work....it leaves the appearance, environment, and interface to you...after all, that's your job...it's what you get paid for.

Early in the beta-testing period of EDatCat, a developer called me and praised the flexibility of EDatCat. "I can design shopping cart around my site...not my my site around my shopping cart," he said. Excellent summation.

EDatCat is designed to give you total control over the look and feel of your web site. How fast you create that is up to you. It is also designed so that you do not have to touch the Perl code. Rather, all of the configuration options and settings are done via the browser-based admin menu. Oh, one more thing....it also doesn't expect you to install it. It does that for you well.

In short, EDatCat is a tool. It's simple enough for the novice to use. As far as speed and ease of use...that depends on you. The first time usually takes a little longer...a few days max. Once you get the hang of it, it's a snap. I've personally set up three sites using EDatCat in the last week...start to finish. Any tool is only as good as the hand that wields it. If you're a skilled craftsman, EDatCat will make you better. If you're just starting out, you'll find that it is easy enough to learn and use effectively...as your skills grow, so will your uses for the software.

All good things take a little work and patience. This has been true since the day the first caveman set foot on the earth who builds a house on the sand knows it better than anyone. Some things are simply a way of life and business. No even the Internet can change that.

Making your mainpage template your home page

Date last updated: 6/10/2002

Using simple tools of trade, you can make your mainpage template the main page.

- Ellison Belt

The whole point of having a home page is to give visitors an easy and common sense entrance to your web site. A good home page will contain easy to read navigational elements, a few featured items or services, fast-loading graphics, and other important in-roads to your online presence- basically- everything a visitor needs to see about your business. Statistically, you have exactly 4.3 seconds to convince your visitors that you are the solution they are seeking...if you are unsuccessful, they will leave and look elsewhere. Creating a good home is your department....I'm not going to focus on what I will focus on is how EDatCat can help.

In your templates directory is a template called "mainpage". This is the template that is called into play whenever the perl script is called directly or the user_action=link&link=mainpage function is used. The purpose of the mainpage template is to provide a simple store entrance. The advantage of using this template as your store entrance is that it contains all of the dynamic elements such as featured products, category links, and others which your clients can maintain from the admin menu without bothering you to update their static index.html page. Using a simple tool of the trade, you can make your mainpage template the default home page of your web site.

As an example, go to <http://www.naturallearning.com> (will open in a new window). This is a site I did for a client who likes to make frequent updates to her site. You will notice that simply going to her home page automatically calls the EDatCat script and displays the mainpage template. Smooth, huh? Now, she can change her featured products and other vital information and have it automatically update to the home page. How did I do that? Simple...

I used a .htaccess file in her root web directory. .htaccess files can be used for a variety of purposes such as setting

requirements for username/password to enter a directory, setting default error messages, setting the rules for how different files will be treated (such as treating a file with the extension .cool as .gif file, displaying it as an image), and, of course, we are discussing here, redirect hits to one page in a different direction.

To accomplish this, I placed a simple text file named ".htaccess" in the root web directory. The file contains a single line in the format of:

```
redirect URL URL-to-go-to
```

So, in other words:

```
redirect /index.html http://www.naturallearning.com/cgi-naturallearning/NATstore.pl
```

This requires that all visitors trying to view the <http://www.naturallearning.com/index.html> page be automatically redirected to the store script...no waits, delays, or meta-refresh tags.

This may not work in all browsers, so I would suggest keeping your index.html file in place with perhaps a link or a meta-refresh tag leading to your store script for the benefit of users who arrive there.

In addition, using redirects- and even meta-refresh tags that refresh too quickly- can play havoc with the process of obtaining search engine listings. A site was recently banned from InfoSeek after submitting a link to a page that was set to redirect. I can't blame InfoSeek here...with the overwhelming amount of irrelevant muck on the Web, I'm always careful to note the URL of a listing on any search engine before I visit. As a general rule, I normally dismiss anything with an AOL Geocities address. Yet, how many times have I clicked a link to something as innocent as a link to a site about raw honey (that's unprocessed for those of you who might wonder) only to wind up at a site entitled "XXX Honey's in the Raw". Hmm...clever use of doorway pages and .htaccess redirects...you have to at least give them that much, however annoying it may be.

The solution to this is to not register this page with search engines. You can also use a robots.txt file to disallow search engine spider-bots from indexing the page you are redirecting.

That about covers it. I'm sure some of you will immediately begin a search for raw honey, so I won't keep you. However, keep the above notes in mind when you get ready to make your mainpage template your actual main page.

For more information, see:

- [Creating Custom Templates](#)

Customer file uploads

Date last updated: 6/7/2004

In the admin utility, select **Program Settings** under **Program Configuration**.

In the **Customer File Uploads** section, enter an upload path and a **Max Upload Size**. The upload path does not have to be the same as your uploads directory. If you like, this can be a directory outside web-accessible space. Be sure to create the directory on your server. The Max Upload Size will be in kilobytes. Only enter the number; do **not** add kb, kB, Kb, KB, etc.

Files uploaded use a filename that consisted of the invoice number (session id), a letter (if there's more than one file uploaded) and the file extension.

At this point, you have a few options for where to add the file upload functionality.

Dedicated Page:

To access the file upload page, use the link function and link to the **fileupload** template. For example,

```
http://www.my-store.com/cgi-bin/ABCstore.pl?user_action=link&link=fileupload
```

If you are linking to this page from elsewhere in an eDatCat-generated page, you should use shortcuts, e.g.

```
%%script_loc%%?user_action=link&link=fileupload
```

Other Page:

In a template of your choosing, you can add the following html to allow a customer to upload a file.

```
<form METHOD="POST" ACTION="%%script_loc%%" ENCTYPE="multipart/form-data" NAME="form1">
<input type="hidden" name="user_action" value="file_upload">
<input type="hidden" name="return_loc" value="%%script_loc%%?user_action=link&link=upload_success">
<input type="file" name="upload_file" size="30">
<input type="submit">
</form>
```

In the above html, the return location is the **upload_success** page. You can also use alternative pages if you like. In addition, you may be able to combine this with JavaScript to add extra functionality.

Custom Programming

Custom Shipping File

Date last updated: 5/26/2004

NOTE: This section is a DRAFT that is included for the benefit of users. It is also subject to the [eDatCat Support Policy](#), in particular Section 3 of the policy.

NOTE: Note that the examples provided below are subject to the same Terms and Conditions as the [EDatCat EUL](#).

In addition to providing several shipping options within the admin utility, EDatCat also allows you to script your own shipping options using a custom shipping script. Using this script, you can write your own perl script to generate your own shipping rates without having to modify EDatCat itself.

- Create a file called customshipping, prefixed by your cart's three-letter dbcode and with your script extension. For example, if your store script is called ABCstore.pl, your custom shipping file would be called ABCcustomshipping.
- If using UNIX, don't forget to use the "shebang" notation to indicate where your perl interpreter is located. For example, if perl is located in the /usr/bin directory, add the following line:

```
#!/usr/bin/perl
```

to the beginning of the script.

- Next, create a function called `get_custom_shipping`. In here, you will generate a new `$shipping_total`, which will be added to the price.
- The final line of the customshipping file must be:

```
1;
```

Below are some examples of custom shipping scripts. You may use these as guides. In addition, EDatCat can write a custom shipping script for you as part of our custom programming service. Contact EDatCat Tech Support for a price quote and a timeframe.

Example 1: Use standard shipping as defined in the EDatCat admin, but free shipping for orders where the final price is greater than or equal to \$100.

```
#!/usr/bin/perl
sub get_custom_shipping {
    if ($tot_price >= 100) {
        $shipping_total = &format_price(0);
    } else {
        $shipping_total = &format_price(&get_shipping());
    }
}
1;
```

Example 2: [To check]

Example 3: Provide free shipping within the state of New York.

```
#!/usr/bin/perl
sub get_custom_shipping {
    if ($shstate eq "NY") {
        $shipping_total = &format_price(0);
    } else {
        $shipping_total = &format_price(&get_shipping());
    }
}
1;
```

Example 4: Provide a shipping "table" (using a fixed price) to states outside North America.

```
#!/usr/bin/perl

# customshipping.cgi for providing a "shipping table" for states in a single country
# outside
# North America.
#
# State which country this applies to:

$intcountry = "ZA"; # assuming South Africa

#
# For each state, enter the line
# $intstate{"JA"} = shipping rate;
```

```

# where JA = state abbreviation
# shipping rate = actual shipping cost
#
# e.g.
#
# $intstate{"JA"} = 2.00;
# State JA would have a set shipping rate of 20% of the total price.

$intstate{"JA"} = 2.00; # shipping charge is $2.00
$intstate{"JB"} = 5.00; # shipping charge is $5.00
$intstate{"JC"} = 0.00; # shipping is free

# etc.

sub get_custom_shipping {
    $shipping_total = &format_price(&get_shipping);
    if ( $shcountry eq $intcountry) {
        $shipping_total = &format_price($intstate{$shstate} * $tot_price);
    }
    return $shipping_total;
}

1; #return true

```

Example 5: Provide a shipping "table" (using a percentage of the total price) to states outside North America.

```

# customshipping.cgi for providing a "shipping table" for states in a single country
outside
# North America.
#
# State which country this applies to:

$intcountry = "ZA"; # Assuming South Africa

#
# For each state, enter the line
# $intstate{"JA"} = shipping rate;
# where JA = state abbreviation
# shipping rate = percentage of total price
#
# e.g.
#
# $intstate{"JA"} = 0.20;
# State JA would have a set shipping rate of 20% of the total price.

$intstate{"JA"} = 0.20; # 20% shipping
$intstate{"JB"} = 0.50; # 50% shipping
$intstate{"JC"} = 0.00; # 0% shipping

# etc.

sub get_custom_shipping {
    $shipping_total = &format_price(&get_shipping);
    if ( $shcountry eq $intcountry) {
        $shipping_total = &format_price($intstate{$shstate} * $tot_price);
    }
    return $shipping_total;
}

```

```
1; #return true
```

Example 6: Provide a shipping "table" based on percentage rather than the default fixed price freight table found in the admin utility.

```
#!/usr/bin/perl
# The following "table" works the same as in the freight table in the admin utility,
# except the final value is a percentage, e.g. 0.20 = 20%.

my @price_ranges =
  ("0;25;0.20",
   "25, 100, 0.15",
   "100, 500, 0.10",
   "500, 1000, 0.05",
   "1000, 999999, 0");

sub get_custom_shipping {
  $shipping_total = &format_price(&get_shipping);

  foreach my($price_line) (@price_ranges) {
    my($from,$to,$frtrate)=split $price_line,///;
    if ($firt_price >=$from and $firt_price<=$to) {
      return $frtrate*$firt_price;
    }
  }
  return $frtrate*$firt_price;
}

1; #return true
```

Changing Accessibility of Admin Functions

Date last updated: 4/27/2004

Version 3.0.5:

(1) If you are using Version 3.0.5 or greater, create a customprogramming script, as described in [this KnowledgeBase article](#).

(2) At the beginning of the script, place the following line:

```
$second_sub{"user action"}="YES";
```

where **user action** is the name of the user action you wish to make accessible to users of the secondary password.

(3) If you are making any changes to the admin templates, you **must** follow [these instructions](#), in order to receive support.

Version 3 - 3.0.4k:

(1) In your admin script (xxxadmin.cgi or xxxadmin.pl), you will find a line:

```
foreach $sub ("modify_data","modify_feat","modify_disc",
```

```
"modify_opt","ord_stat","acct_stat","welcome_msg","mail_list","gen_indexes","regen_indexes",  
"sls_rpts","utilites","call_purge","exp_payinfo","store_disc","inv_report","util_backup") {#7/26/02  
$second_sub{$sub}="YES"}
```

Modify this line as follows:

```
foreach $sub ("user action", "modify_data","modify_feat","modify_disc",  
"modify_opt","ord_stat","acct_stat","welcome_msg","mail_list","gen_indexes","regen_indexes",  
"sls_rpts","utilites","call_purge","exp_payinfo","store_disc","inv_report","util_backup") {#7/26/02  
$second_sub{$sub}="YES"}
```

where **user action** is the name of the user action you wish to make accessible to users of the secondary password.

(3) If you are making any changes to the admin templates, you **must** follow [these instructions](#), in order to receive support.

Notes:

(i) We do not provide support for scripts customized by anybody other than eDatCat staff. In addition, we reserve the right to refuse support to people who have any portion of their scripts modified by anybody other than eDatCat staff. (This has always been eDatCat's policy. We cannot guarantee support if you modify the scripts. In particular, if you report a problem that is caused by your customizations, we reserve the right to bill you.)

(ii) If you do not follow the instructions referred to in step (3), we will bill you for support. This policy is outlined in the `README.txt` file in the admin directory.