

Volume

1

PATHFINDERS SOFTWARE

Shopping Cart System

Programmers Reference Guide

LAPD SHOPPING CART

Programmers Reference Guide

© Herbert Rubin
701 S. Grant St
San Mateo, CA 94402
Phone 650-343-4571 • Fax 650-343-4675
Document Version 1.08 June 5, 2007

Table of Contents

Session Tracking.....	1
How PHP Uses The Session.....	1
How D3 Uses The Session	3
index.php revealed	4
A Note About Order Forms	4
GETCARTSUMM.SUB.....	4
ADDCARTSKU.SUB	4
GETCATEGORY.SUB	5
GETCATNAME.SUB.....	5
GETPRODLIST.SUB	5
SEARCHPRODLIST.SUB.....	6
Breadcrumbs.....	6
GETTABS.SUB.....	6
GETSYSTEMINFO.SUB	6
login.php revealed	7
CHECKPASSWORD.SUB	7
GETSECURITY.SUB	7
GETPROFILE.SUB	8
LOGOFF.SUB.....	8
forgot_login.php revealed	8
GETPROFILE.SUB	8
CHECKPASSWORD.SUB	8
view_cart.php revealed	9
ADDCARTSKU.SUB	9
REMOVECARTSKU.SUB.....	9
SAVECART.SUB	9
GETCARTHEADER.SUB.....	9
GETCARTDETAIL.SUB	10
product.php revealed.....	11
Product Images	11
GETPRODUCTINFO.SUB	11

Bullets:	11
Price Chart:	12
GETBREADBYSKU.SUB	12
my_profile.php revealed	13
GETPROFILE.SUB	13
SAVEPROFILE.SUB	13
STILLLOGGEDIN.SUB	13
checkout.php revealed	14
STILLLOGGEDIN.SUB	14
GETSECURITY.SUB	14
GETSUBACCTLIST.SUB	14
GETBILLTO.SUB	14
GETSHIPTO.SUB	15
SETCARTHEADER.SUB	15
GETCARTHEADER.SUB	15
GETCARTDETAIL.SUB	15
GETCARTSUMM.SUB	15
checkout2.php revealed	16
STILLLOGGEDIN.SUB	16
GETSECURITY.SUB	16
GETCARTHEADER.SUB	16
GETCARTDETAIL.SUB	16
GETBILLTO.SUB	16
GETSHIPMETH.SUB	17
SETCARTHEADER.SUB	17
GETCARTHEADER.SUB	17
GETCARTDETAIL.SUB	17
SUBMITORDER.SUB	17
checkout3.php revealed	18
GETPAYMETHODS.SUB	18
GETCCLIST.SUB	18
SETPAYMENTCHOICE.SUB	18
STILLLOGGEDIN.SUB	18
GETSECURITY.SUB	18
checkout4.php revealed	19
STILLLOGGEDIN.SUB	19
GETSECURITY.SUB	19
GETBILLTO.SUB	19
GETCARTHEADER.SUB	19

GETCARTDETAIL.SUB	19
SUBMITORDER.SUB	19
GETPAYMETHODS.SUB	19
GETCCLIST.SUB	19
GETPAYMENTCHOICE.SUB	20
checkout5.php revealed	21
GETORDERHEADER.SUB	21
GETORDERDETAIL.SUB.....	21
STILLLOGGEDIN.SUB	21
GETSECURITY.SUB	21
invoice_history.php revealed	22
GETINVLIST.SUB.....	22
COUNTINVLIST.SUB	22
EMAILORDERS.SUB.....	22
FAXORDERS.SUB	23
Appendix A: PHP to D3 Subroutine List	24
Appendix B: Approval System.....	26
ORGUNITS file	26
USERS file	26
GETORGUNITINFO.SUB(orgid).....	26
GETORGUSERS.SUB(orgunitid).....	28
CREATEORGUNIT.SUB(orgunitid)	28
GETUSERLIST.SUB(orgunitid).....	28
UPDATEORGUNIT.SUB(orgunitid)	28
CREATEUSER.SUB(orgunitid)	28
Index	29

Session Tracking

Both PHP and D3 track your session.

When a user surfs over to the shopping cart, a cookie with a session number is automatically set on his browser. This random session number is used in two places.

How PHP Uses The Session

The session number is the name of a file in Linux. Inside this file are kept variables that are persistent across web pages (the session). These include strings, arrays, integers, etc..

The main point here is that if a user logs in, we have a lot more information to track that if he is just a guest or casual surfer. Here is a current list of PHP session variables of a casual web surfer:

```
[progversion] => 1.0.0
[loggedin] =>
[fullname] =>
[userid] =>
[userinfo] =>
[security] => Array ( )
[referrer] =>
[cart_refer_link] =>
[cart_refer_display] =>
[breadcrumbs] => Array ( )
[last_order_id] =>
[last_order_type] =>
[billto_num] =>
[billto_name] =>
[language] => C
```

Notice how some of the variables are empty arrays.

BASIC CONCEPTS

Here is a list of the same session variables after a person is done logging in:

```
[progversion] => 1.0.0
[loggedin] => 1
[fullname] =>
[userid] => herb
[userinfo] => Array (
    [ID] => herb
    [TYPE] => C
    [NUMBER] => 123
    [NAME] => JEFF TEST CUSTOMER
    [ADDRESS1] => 4471 N. BUNGALOW LANE
    [ADDRESS2] =>
    [CITY] => FRESNO
    [STATE] => CA
    [ZIP] => 93704
    [SECQUESTION] =>
    [SECANSWER] =>
    [WEBSITE] => www.laner.com
    [EMAIL] => keydat@keydat.com
    [CREDITEMAIL] => jkaufman@keydat.com
    [TYPECODE] => C
)
[security] => Array (
    [WEBCART] => Y
    [EDITPROFILE] => Y
    [SUBACCOUNTS] => Y
    [MASTERACCOUNT] => Y
    [EDITSUBACCOUNTS] => Y
    [ORDERSTATALL] => Y
    [INVENTORYINQ] => Y
    [FORMCOUNT] => 0
    [FORMONLY] => N
    [place_orders] => Y
    [show_inventory] => Y
    [show_my_profile] => Y
    [show_administration] => Y
    [show_account_status] => Y
    [show_order_status] => Y
    [show_invoice_history] => Y
    [show_payment_history] => Y
    [show_last_sale_history] => Y
    [show_purchase_history] => Y
    [show_quote_inquiry] => Y
    [show_customer_inquiry] => Y
    [show_search_customer] => N
    [show_saved_carts] => Y
    [change_price] => N
    [create_new_account] => N
)

[referrer] =>
[cart_refer_link] =>
[cart_refer_display] =>
[breadcrumbs] => Array ( )
```

BASIC CONCEPTS

```
[last_order_id] =>  
[last_order_type] =>  
[billto_num] => 123  
[billto_name] => JEFF TEST CUSTOMER  
[language] => C
```

How D3 Uses The Session

In D3, there is a file called WEB-ACTIVITY that is used to store session information. It uses the same session id stored in the cookie as the item-id of the D3 record. See the program WAS.BP SR.WAS.SESSION.REC

```
A<1> = date stamp  
A<2> = time stamp
```

An example of a typical session id is “f1557a2ccaf1474f773744fb6f2075f9” which as you can see is very random.

index.php revealed

Displaying categories and ordering products happens mostly on index.php.

A Note About Order Forms

Some users will be restricted to shopping from order forms. We will know this after they logon. This means they cannot shop from categories. They will only be allowed to see one or more prebuilt order forms. If the current user is one of these, he will be redirected to orderforms.php and not be allowed to stay on index.php.

GETCARTSUMM.SUB

On the left side we display a quick count of what's in the shopping cart.

```
[COUNT] => 0
[PRICE] => 0.00
[WEIGHT] => 0.00
```

COUNT is the number of products in the cart. PRICE is the total cost. WEIGHT is the total weight of all items added together.

ADDCARTSKU.SUB

After updating quantities (and prices if you are a salesman), you will click the submit button to add these items to your shopping cart. Optionally, you can simply press enter while your mouse is still inside a quantity box and that will still submit the page. This subroutine is also called on view_cart.php and saved_carts2.php.

```
if ($_SESSION['security']['change_price'] == "Y") {
    add_cart_items($items, $cartname, $prices);
} else {
    add_cart_items($items, $cartname);
}
```

Data returned to D3 will be in one of two forms:

```
[cartname] =>
[12158] => 1
[60158] => 3
[12159] => 2
[60159] => 1
```

A blank "cartname" means main shopping cart not a saved cart. The other form is:

```
[cartname] =>
[12158] => 1|9.49
[60158] => 3|11.99
[12159] => 2|32.50
[60159] => 1|129.95
```

Where the pipe symbol separates the quantity from the price.

GETCATEGORY.SUB

This subroutine actually returns a list of subcategories. It is misleading. However, it will pass in a category code and expect back a list of the next lower level categories. If the passed in category is blank, then return back the top level categories. Each attribute returned is this kind of data:

```
VM  Type of data
--  -----
1 = category code number
2 = category display name
3 = parent category code number
4 = number of child categories
5 = number of products in the category
6 = image id (not used at this time)
```

GETCATNAME.SUB

This subroutine returns information about the current category requested.

```
[catno]          => 01
[catname]        => Chemicals
[parentcatno]    => D4
[prodcount]      => 4
[mfg_logos]      => may be multivalued (optional)
[prod_logos]     => may be multivalued (optional)
[bullets]        => may be multivalued (optional)
```

The last three are displayed on the page below the child category box, but above the product list.

GETPRODLIST.SUB

Returns all the products in a category with pagination. This means that PHP can request a specific page and a number of products per page.

The first and second attribute returned are special and the 3rd and up are product information.

```
First attribute = column header names
Second attribute = justification of columns: L, C or R
third attribute and up are raw product data
```

Note: Certain column names trigger a special effect in PHP:

```
Qty to Order      Input box
Your Price or     Input box if salesman
Sku#              Clickable link to product.php page
```

SEARCHPRODLIST.SUB

Returns similar data as GETPRODLIST.SUB except the first attribute is the product count.

First attribute	Number of products that match search
Second attribute	Column header names
Third attribute	Justification of columns: L, C or R
Fourth attribute +	Raw product data

We need this because we don't have the luxury of knowing how many products are in the specific category. Note: Certain column names trigger a special effect in PHP:

Qty to Order Input box
Your Price or Input box if salesman
Sku# Clickable link to product.php page

Breadcrumbs

Breadcrumbs are text links like this:

Home > Plumbing Supplies > Check Valves and Tubing > Air Check Valves

They are self managed by PHP. There is no interaction in D3 other than the category subroutines.

GETTABS.SUB

This function returns the top tabs that show on every page. It returns a keyed array.

```
[HOME]                => ./index.php  
[SHOPPING CART]     => ./view_cart.php  
[MY ACCOUNT]        => ./my_profile.php  
[CONTACT US]        => ./contact_us.php
```

GETSYSTEMINFO.SUB

This function returns some specific variables that the website uses globally.

```
[companyname]        => Western Pacific Supply Test Site  
[contact_email]     => jkaufman@keydata.us  
[contact_phone]     => 510/486-9015  
[contact_name]      => Jeffrey Kaufman  
[theme]             => blue  
[order_status_all]  => 0  
[logo]              => /images/WPS_headerB.gif
```

Note: order_status_all is simply whether you want to show a dropdown box in order history or instead hide the dropdown box and always show all invoices. Some people want to view only unpaid ones so the dropdown box is a nice filter. 0 = dropdown box, 1 = hide dropdown box

login.php revealed

Users are allowed to shop as a guest user and not login. However, they must login to checkout. After login, their cart is preserved under their proper billto code. If they don't have an account, we do allow them to fill out a form to get one. (new_customer.php).

CHECKPASSWORD.SUB

When a user submits the login form, PHP calls this D3 function. If the userid and password combination matches, let the user login. If the password is blank and the secure answer is not then compare it to the secure question. If it matches, then allow them to login. The returned data is either 1 or 0. 1 means success. 0 means failed to login.

GETSECURITY.SUB

This function is called right after the password is valid to establish a number of yes/no parameters. It is also called on a regular basis on other PHP pages. These settings affect the user's experience on the website for this session. They are stored in the PHP session.

```
[WEBCART] => Y
[EDITPROFILE] => Y
[SUBACCOUNTS] => Y
[MASTERACCOUNT] => Y
[EDITSUBACCOUNTS] => Y
[ORDERSTATALL] => Y
[INVENTORYINQ] => Y
[FORMCOUNT] => 0
[FORMONLY] => N
[place_orders] => Y
[show_inventory] => Y
[show_my_profile] => Y
[show_administration] => Y
[show_account_status] => Y
[show_order_status] => Y
[show_invoice_history] => Y
[show_payment_history] => Y
[show_last_sale_history] => Y
[show_purchase_history] => Y
[show_quote_inquiry] => Y
[show_customer_inquiry] => Y
[show_search_customer] => N
[show_saved_carts] => Y
[change_price] => N
[create_new_account] => N
```

GETPROFILE.SUB

This function is also called after the user successfully logs in. It returns data like this to be stored in the PHP session:

```
[ID] => herb
[TYPE] => C
[NUMBER] => 123
[NAME] => JEFF TEST CUSTOMER
[ADDRESS1] => 4471 N. BUNGALOW LANE
[ADDRESS2] =>
[CITY] => FRESNO
[STATE] => CA
[ZIP] => 93704
[SECQUESTION] =>
[SECANSWER] =>
[WEBSITE] => www.laner.com
[EMAIL] => keydat@keydat.com
[CREDITEMAIL] => jkaufman@keydat.com
[TYPECODE] => C
```

This is also called on the my_profile.php page.

LOGOFF.SUB

This function can be called from any page on the site but is related to login so we describe it here. Both D3 session and PHP session information are affected by logging off.

forgot_login.php revealed

This page helps the login.page if a user forgets his password. He may still be able to login by using his secure question and answer.

GETPROFILE.SUB

This function returns some data including the SECQUESTION to be displayed in PHP.

CHECKPASSWORD.SUB

When a user submits the forgot_login form, PHP calls this D3 function. The password will be blank and the secure answer will not be. If it matches, then allow them to login. The returned data is either 1 or 0. 1 means success. 0 means failed to login.

However, if the SECANSWER is blank and the response from the user is also blank. Don't allow that to be a successful login.

view_cart.php revealed

This page is used to show what is in your shopping cart already. It is not used to view contents of saved shopping carts. Note: The user does not have to be logged to get to this page.

ADDCARTSKU.SUB

After updating quantities (and prices if you are a salesman), you will click the submit button to update these items in your shopping cart. Optionally, you can simply press enter while your mouse is still inside a quantity box and that will still submit the page. This subroutine is also called on index.php and saved_carts2.php

REMOVECARTSKU.SUB

This subroutine removes items from a shopping cart. Data passed into D3 looks like this:

```
[cartname] =>
[12158]     => D
[60158]     => D
[12159]     => D
[60159]     => D
```

A blank “cartname” means main shopping cart not a saved cart. This subroutine is also called in saved_carts2.php.

SAVECART.SUB

This subroutine tells D3 to save the current contents of the main shopping cart as a saved shopping cart. Data passed into D3 looks like this:

```
[cartname] => mycart1
```

Only letters, numbers, spaces, underscores, dots and dashes are allowed in the name of the saved cart. This is checked by Javascript in the browser.

GETCARTHEADER.SUB

Return a set of variables about the shopping cart. Input is:

```
[cartname] => mycart1
```

A blank “cartname” means main shopping cart not a saved cart. Data returned is:

```
[COUNT]   => 4
[PRICE]    => 5,554.09
[WEIGHT]   => 85.22
```

You might notice that this data is similar to GETCARTSUMM.SUB. It is ☺

GETCARTDETAIL.SUB

Return all the products in the shopping cart. Only one field is passed in:

```
[cartname] => mycart1
```

A blank “cartname” means main shopping cart not a “saved cart”. The data returned looks like this:

```
[0] => Array (
    [0] => Sku#
    [1] => Description
    [2] => Qty Ordered
    [3] => Unit
    [4] => Price
    [5] => Per
    [6] => Extended
    [7] =>
)

[1] => Array (
    [0] => C
    [1] => L
    [2] => C
    [3] => C
    [4] => R
    [5] => C
    [6] => R
    [7] =>
)

[2] => Array (
    [0] => 60158
    [1] => 2101 Unibody Gate Valve 1-1/2 inch Slip x
Slip PVC Wht 16 lbs. per carton of 30
    [2] => 6
    [3] => EA
    [4] => 8.10
    [5] => EA
    [6] => 48.60
    [7] =>
)
```

Notice that the first 2 rows of data are special and the 3rd row is the first product.

Sku#	Clickable link to product.php page
Qty to Order	Input box
Your Price or	Input box if salesman

Its similar to GETPRODLIST.SUB.

product.php revealed

This page shows the details of one specific product. It will reveal the full breadcrumb path of categories.

Product Images

We allow for 2 rows of images. The first row we call the manufacturers logos and the second row we call the product images.

GETPRODUCTINFO.SUB

This function returns 15 fields describing the product sku.

Field Number	Field Name	Notes
1	SKU	
2	Heading	
3	MFG Logos	Multivalued
4	Product Images	Multivalued
5	Bullets	Multivalued
6	Description	Put in tags if you want to control line wrapping.
7	MFG Part	
8	UPC	
9	Stock Type	
10	Stock Units	
11	List Price	Hidden column if blank
12	Your Price	
13	Units	
14	Price Chart	Multivalued
15	Categories	

Bullets:

We allow for optional bullet points below the optional images. Here is a sample of returned multivalued data in field 5.

```
[0] => Plastic Paddles
[1] => All 2 inch Slips are 2 inch x 2-1/2 inch
[2] => Slip=Female Glue
[3] => Spig=Male Glue
[4] => Fipt=Female Pipe Thread
[5] => Mipt=Male Pipe Thread
```

Price Chart:

We allow for a price chart for quantity discounts, etc. If this chart is missing then we do not show an input box for ordering.

```
[0] => EACH (1)
[1] => 5.83
```

GETBREADBYSKU.SUB

This function returns the list of categories to drill down to this product. The input data is **skunum**. The returned data looks like this:

```
Array (
  [0] => Array (
    [0] => D1
    [1] => Gate Valves
  )
  [1] => Array (
    [0] => D1_GVP
    [1] => Plastic Gate Valves
  )
  [2] => Array (
    [0] => D1_GVP_GV01
    [1] => 1-1/2" Plastic Gate Valves
  )
)
```

Make sure the first entry is the most general category and the last entry is the most specific category. So, the breadcrumb line would display like this:

Home > Gate Valves > Plastic Gate Valves > 1-1/2" Plastic Gate Valves

my_profile.php revealed

This page is where a user can change his password, security question/answer and email address.

GETPROFILE.SUB

This function is also called after the user successfully logs in on the login.php page. It returns data like this to be stored in the PHP session:

```
[ID] => herb
[TYPE] => C
[NUMBER] => 123
[NAME] => JEFF TEST CUSTOMER
[ADDRESS1] => 4471 N. BUNGALOW LANE
[ADDRESS2] =>
[CITY] => FRESNO
[STATE] => CA
[ZIP] => 93704
[SECQUESTION] =>
[SECANSWER] =>
[WEBSITE] => www.laner.com
[EMAIL] => keydat@keydat.com
[CREDITEMAIL] => jkaufman@keydat.com
[TYPECODE] => C
```

SAVEPROFILE.SUB

This function is called when the user presses the Submit button. This data is passed to D3:

```
[PASSWORD] => test
[PASSWORD2] => test
[SECQUESTION] => King?
[SECANSWER] => Burger
[EMAIL] => keydat@keydat.com
[CREDITEMAIL] => jkaufman@keydat.com
[WEBSITE] => www.laner.com
```

So, just save it to the users record that owns this session. Errors returned back will be displayed on the page. The email addresses are checked for validity in Javascript.

STILLOGGEDIN.SUB

If not logged in, the user is redirect to the login page. In most cases, we remember where he wanted to go by storing the URL in the PHP session. Then, after a successful login, we redirect him to it. This function is called on many pages.

If null or 0 (zero) is returned, that means logged out. Normally, return a 1 (one) if the user is still logged in.

checkout.php revealed

This is the first page that begins the checkout process. A user must be logged in or else they get redirected to login.php. The most important subroutine is SETCARTHEADER.SUB.

STILLOGGEDIN.SUB

If not logged in, the user is redirect to the login page. In most cases, we remember where he wanted to go by storing the URL in the PHP session. Then, after a successful login, we redirect him to it. This function is called on many pages.

If null or 0 (zero) is returned, that means logged out. Normally, return a 1 (one) if the user is still logged in.

GETSECURITY.SUB

If still logged in, refresh the security settings.

GETSUBACCTLIST.SUB

Returns a list of open or all subaccounts (shipto's).

Input:

```
kind=A (all)
kind=O (open)
kind=I (inactive)
```

It returns a keyed array with the account numbers and account names of all the shipto's.

GETBILLTO.SUB

Input **custno**, returns an array of address information. If **custno** is blank then return this user's default billto.

```
Array (
  [NUMBER] => 123
  [NAME] => JEFF TEST CUSTOMER
  [ADDRESS1] => 4471 N. BUNGALOW LANE
  [ADDRESS2] =>
  [CITY] => FRESNO
  [STATE] => CA
  [ZIP] => 93704
  [OPENACCT] => N
  [PHONE] => 559/432-3832
  [FAX] => 448-8834
  [GUEST] => N
)
```

The GUEST variable is to help us decide if a salesman needs to specify which client he is shopping for. Its only Y or N.

GETSHIPTO.SUB

Input custno, returns an array of address information.

SETCARTHEADER.SUB

The most important subroutine on this page. It submits data to the cart header record such as ship to address information. It should only update or replace variables that are submitted, not a wipeout of all existing data.

It returns an error array to be displayed if the data is not valid or missing required fields. If there are no errors, the browser is redirected to checkout2.php.

GETCARTHEADER.SUB

Return information about the cart, just like on view_cart.php

GETCARTDETAIL.SUB

Return products in cart, just like on view_cart.php.

GETCARTSUMM.SUB

This is called on almost all pages.

checkout2.php revealed

Once the user has selected his billto and shipto addresses properly, the browser is redirected to checkout2.php. This page selects his shipping options.

Also, if the order is to become a quote, it can be submitted here. We don't have to collect payment information.

STILLOGGEDIN.SUB

If not logged in, the user is redirect to the login page. In most cases, we remember where he wanted to go by storing the URL in the PHP session. Then, after a successful login, we redirect him to it. This function is called on many pages.

If null or 0 (zero) is returned, that means logged out. Normally, return a 1 (one) if the user is still logged in.

GETSECURITY.SUB

If still logged in, refresh the security settings.

GETCARTHEADER.SUB

Return information about the cart, just like on view_cart.php

If the field NEXT = CC then proceed to checkout3.php. Credit card required.

If the field NEXT = CONFIRM then jump to checkout4.php. Pay later.

GETCARTDETAIL.SUB

Return products in cart, just like on view_cart.php.

GETBILLTO.SUB

Input **custno**, returns an array of address information. If **custno** is blank then return this user's default billto.

```
Array (  
    [NUMBER] => 123  
    [NAME] => JEFF TEST CUSTOMER  
    [ADDRESS1] => 4471 N. BUNGALOW LANE  
    [ADDRESS2] =>  
    [CITY] => FRESNO  
    [STATE] => CA  
    [ZIP] => 93704  
    [OPENACCT] => N  
    [PHONE] => 559/432-3832  
    [FAX] => 448-8834  
    [GUEST] => N  
)
```

The GUEST variable is to help us decide if a salesman needs to specify which client he is shopping for. Its only Y or N.

GETSHIPMETH.SUB

Return a list of shipping methods: code vm description vm charges

```
Array (
  [0] => Array (
    [0] => UPS
    [1] => UPS GROUND
    [2] => $5.00
  )
  [1] => Array (
    [0] => UPSB
    [1] => UPS BLUE
    [2] => $20.00
  )
  [2] => Array (
    [0] => UPSO
    [1] => UPS ORANGE
    [2] => $10.00
  )
)
```

SETCARTHEADER.SUB

It submits data to the cart header record. It only sets SHIPMETHOD.

It should only update or replace variables that are submitted, not a wipeout of all existing data.

GETCARTHEADER.SUB

Return information about the cart, just like on view_cart.php

GETCARTDETAIL.SUB

Return products in cart, just like on view_cart.php.

SUBMITORDER.SUB

Tells the database to process the order or quote. It inputs one variable **rectype**..

```
rectype = Q (submit quote)
```

```
rectype = O (submit order)
```

checkout3.php revealed

This page is for entering credit card information. You may have existing credit cards on file that you may want to use.

GETPAYMETHODS.SUB

Return a list of credit card types that we accept.

```
Array (
  [0] => Array (
    [0] => AE
    [1] => AMERICAN EXPRESS
    [2] => CC
  )
  [1] => Array (
    [0] => D
    [1] => DISCOVER
    [2] => CC
  )
  [2] => Array (
    [0] => MC
    [1] => MASTER CARD
    [2] => CC
  )
  [3] => Array (
    [0] => V
    [1] => VISA
    [2] => CC
  )
)
```

GETCCLIST.SUB

Return a list of existing credit card on file for this client. Foresite only stores one at this time.

SETPAYMENTCHOICE.SUB

Save order payment method as chosen by user.

If paychoice = "N" then expect lots of credit card data also submitted.

If paychoice is a number then use that existing credit card number

Return an array of errors. If no errors the browser is redirected to checkout4.php for final confirmation.

STILLOGGEDIN.SUB

If not logged in, the user is redirect to the login page.

GETSECURITY.SUB

If still logged in, refresh the security settings.

checkout4.php revealed

This page is the final confirmation page. The user is just reviewing the order. Once submitted, the most important subroutine is `SUBMITORDER.SUB`.

STILLOGGEDIN.SUB

If not logged in, the user is redirect to the login page. In most cases, we remember where he wanted to go by storing the URL in the PHP session. Then, after a successful login, we redirect him to it. This function is called on many pages.

If null or 0 (zero) is returned, that means logged out. Normally, return a 1 (one) if the user is still logged in.

GETSECURITY.SUB

If still logged in, refresh the security settings.

GETBILLTO.SUB

Input `custno`, returns an array of address information. If `custno` is blank then return this user's default billto.

GETCARTHEADER.SUB

Return information about the cart, just like on `view_cart.php`

GETCARTDETAIL.SUB

Return products in cart, just like on `view_cart.php`.

SUBMITORDER.SUB

Tells the database to process the order or quote. It inputs one variable `rectype`..

```
rectype = Q (submit quote)
rectype = O (submit order)
```

Returns:

```
RECTYPE = O or Q
RECNUMBER = order number
```

This returned data is used on `checkout5.php` and `checkout6.php`

GETPAYMETHODS.SUB

Return a list of credit card types that we accept.

GETCCLIST.SUB

Return a list of existing credit card on file for this client. Foresite only stores one at this time.

GETPAYMENTCHOICE.SUB

Return the order payment method as chosen by the user.

```
Array (  
  [paychoice] =>  
  [type] =>  
  [number] =>  
  [expmonth] =>  
  [expyear] =>  
  [name] =>  
  [address1] =>  
  [address2] =>  
  [city] =>  
  [state] =>  
  [zip] =>  
  [phone] =>  
  [cvv2] =>  
  [saveinfo] =>  
  [numberxx] =>  
)
```

However if `carthead[“NEXT”]` = “CONFIRM” then we just display “Open” Account.

checkout5.php revealed

Displays the receipt page (thank you page) of an order. Since some orders are put on hold we can't just call the get invoice header subroutine. This page is mostly the same for checkout6.php.

GETORDERHEADER.SUB

Return information about the order. In the MEMO field the pipe symbol “|” is converted to html <p> paragraph marks by PHP.

GETORDERDETAIL.SUB

Return information about the order.

STILLOGGEDIN.SUB

If not logged in, the user is redirect to the login page. In most cases, we remember where he wanted to go by storing the URL in the PHP session. Then, after a successful login, we redirect him to it. This function is called on many pages.

If null or 0 (zero) is returned, that means logged out. Normally, return a 1 (one) if the user is still logged in.

GETSECURITY.SUB

If still logged in, refresh the security settings.

invoice_history.php revealed

Displays information about past invoices. There is a separate page for order status.

GETINVLIST.SUB

Returns an array of invoices. (Fillarray). Fields passed in are:

Kind, sortby, page, perpage, search, dir

kind	A=all, O=unpaid/open
sortby	IN=inv#, BR=branch, SH=shipto, OD=order date, ID=inv date, PO=po, OB=ordered by, JO=joborder, IT=inv total
page	Page number
perpage	Lines per page
search	Search string if any. Match against any field displayed
dir	A = ascending sort B = descending sort

COUNTINVLIST.SUB

Return the number of invoices found without pagination. Either all or by search criteria. This is used to decide if we even need pagination.

sortby	IN=inv#, BR=branch, SH=shipto, OD=order date, ID=inv date, PO=po, OB=ordered by, JO=joborder, IT=inv total
search	Search string if any. Match against any field displayed
dir	A = ascending sort B = descending sort

EMAILORDERS.SUB

Passes "email" in the form for the email address to send invoices to. Also, in the form will be numeric fields where each value is an invoice number to be included in the email.

FAXORDERS.SUB

Passes "email" in the form for the email address to send invoices to. Also, in the form will be numeric fields where each value is an invoice number to be included in the email.

Appendix A: PHP to D3 Subroutine List

The following is a quick list of the D3 subroutines this PHP shopping cart calls directly.

ADDCARTSKU.SUB
CHECKPASSWORD.SUB
COUNTINVHIST.SUB
COUNTINVLINES.SUB
COUNTINVLIST.SUB
COUNTLSHIST.SUB
COUNTORDHIST.SUB
COUNTORDLINES.SUB
COUNTPAYHIST.SUB
COUNTQUOHIST.SUB
CREATEACCOUNTOK.SUB
CREATEPROFILE.SUB
DELETECART.SUB
EMAILORDERS.SUB
EMAILQUOTES.SUB
FAXORDERS.SUB
FAXQUOTES.SUB
GET.VENDORS.SUB
GETBILLTO.SUB
GETBILLTOTREE.SUB
GETBRANCHDETAIL.SUB
GETBRANCHLIST.SUB
GETBRANCHTREE.SUB
GETBREADBYSKU.SUB
GETCARTDETAIL.SUB
GETCARTHEADER.SUB
GETCARTLIST.SUB
GETCARTSUMM.SUB
GETCATEGORY.SUB
GETCATNAME.SUB
GETCCLIST.SUB
GETCHECKOUTSHIPTO.SUB
GETCUSTPURCHDET.SUB
GETCUSTPURCHSUMM.SUB
GETFORMLIST.SUB
GETFORMPRODLIST.SUB
GETINVAGING.SUB
GETINVDETAIL.SUB
GETINVHEADER.SUB
GETINVHIST.SUB

GETINVLINES.SUB
GETINVLIST.SUB
GETLSHIST.SUB
GETORDERDETAIL.SUB
GETORDERHEADER.SUB
GETORDHIST.SUB
GETORDLINES.SUB
GETPAYDETAIL.SUB
GETPAYHEADER.SUB
GETPAYHIST.SUB
GETPAYMENTCHOICE.SUB
GETPAYMETHODS.SUB
GETPRODLIST.SUB
GETPRODUCTINFO.SUB
GETPROFILE.SUB
GETQUODETAIL.SUB
GETQUOHEADER.SUB
GETQUOHIST.SUB
GETSECURITY.SUB
GETSHIPMETH.SUB
GETSHIPTO.SUB
GETSUBACCTLIST.SUB
GETSUBACCTSUM.SUB
GETSUBPROFILE.SUB
LOGOFF.SUB
REMOVECARTSKU.SUB
SAVECART.SUB
SAVEPROFILE.SUB
SAVESUBPROFILE.SUB
SEARCHBILLTO.SUB
SEARCHPRODLIST.SUB
SETBILLTO.SUB
SETCARTHEADER.SUB
SETCHECKOUTSHIPTO.SUB
SETPAYMENTCHOICE.SUB
STILLOGGEDIN.SUB
SUBMITORDER.SUB
UNFORMATPHONE.SUB
USECART.SUB

Appendix B: Approval System

Notes about the web ordering approval system. For those clients that place orders on hold till their managers approve them.

ORGUNITS file

A0 = ORG Item-ID
A1 = Master Billto Code
A2 = Billto Code
A3 = Shipto Code, Taxable
A4 = Shipto Code, Non-Taxable
A5 = Display Name
A6 = Parent Item-ID
A7 = Children Item-IDs (mv)
A8 = Budget Per Order, Taxable
A9 = Budget Per Month, Taxable
A10 = User IDs (mv)
A11 = Approval Person Y or N (mv)
A12 = Ordering Person Y or N (mv)
A13 = Budget Per Order, Non-Taxable
A14 = Budget Per Month, Non-Taxable

USERS file

A0 = User Item-ID
A3 = Password
A22 = Full Name
A25 = Budget Per Order taxable
A26 = Budget Per Order non-taxable
A29 = Email

GETORGUNITINFO.SUB(orgid)

Returns a list of existing organizational units, similarly how Categories works.

```
GETORGUNITINFO ("")
```

Would return the top level org units like this:

```
Array (  
  [0] => Array (  
    [0] => WC  
    [1] => West Coast Territory  
    [2] => 3  
    [3] =>  
    [4] => 17  
  )  
  [1] => Array (  
    [0] => WC  
    [1] => West Coast Territory  
    [2] => 3  
    [3] =>  
    [4] => 17  
  )  
)
```

```

        [0] => NYC
        [1] => New York Metropolitan Area
        [2] => 12
        [3] =>
        [4] => 9
    )
[2] => Array (
    [0] => SE
    [1] => Southeastern US
    [2] => 8
    [3] =>
    [4] => 29
)
)

```

Note:

Field [2] is the **child count**. In this case, SE (Southeastern US) has 8 Branches.

Field[3] is the **parent code**. In this case there is none. These are top level org units.

Field[4] is the **user count**. In this case WC has 17 users in it. Some are approval users, some are ordering users, some are both.

```
GETORGUNITINFO ("WC")
```

Might return all the west coast branches:

```

Array (
  [0] => Array (
    [0] => WASHOR
    [1] => Washington and Oregon
    [2] => 9
    [3] => WC
    [4] => 3
  )
  [1] => Array (
    [0] => CANORTH
    [1] => Northern California
    [2] => 7
    [3] => WC
    [4] => 9
  )
  [2] => Array (
    [0] => CASOUTH
    [1] => Southern California
    [2] => 15
    [3] => WC
    [4] => 5
  )
)

```

Notice the last one: CASOUTH is a branch and its parent is WC (the West Coast District). And it has 15 Offices.

GETORGUSERS.SUB(orgunitid)

Returns a list of existing users in this organizational unit. With a code denoting if this user is Approval, Ordering or Both.

```
[userid] = JSMITH  
[name]   = John Smith  
[allowed] = B  
[budget_tax] = 1500  
[budget_nontax] = 1600  
[budget_monthly_tax] = 5000  
[budget_monthly_nontax] = 6000
```

[allowed] can be either: A, O or B. Approval person, Ordering person or Both.

CREATEORGUNIT.SUB(orgunitid)

Create a location record in ORGCHART file.

GETUSERLIST.SUB(orgunitid)

Return a list of users that are associated with a specific location.

UPDATEORGUNIT.SUB(orgunitid)

Update a location record in ORGCHART with new fields.

CREATEUSER.SUB(orgunitid)

Create a web user and associate it to its first ORG CHART location.

Index

Breadcrumbs, 6
cookie, 1, 3

guest, 1, 7
WEB-ACTIVITY, 3