



# **VB.NET Functions (Quick Guide)**

For use with all Asp.Net versions of PDshop  
Revised: 9/1/2012

PageDown Technology, LLC / Copyright 2002-2012 All Rights Reserved.

# Table of Contents

Table of Contents .....	2
Getting Started .....	3
PDshop Namespaces.....	3
PDshop Functions Class.....	3
Sample ASPX script .....	4
Sample Master Page .....	5
Customer Functions .....	6
getCustomerVariables.....	6
Order Functions .....	7
getOrderVariables .....	7
Email Functions.....	9
SendEmail.....	9
Database Operations – Show data in your .aspx page.....	10
BindThisSQL.....	10
Database Operations – Run/Execute any SQL .....	11
SQLExecute.....	11
Database Operations - Read .....	12
OpenDataReader.....	12
ReadData .....	12
ReadData (other variations) .....	12
CloseData .....	12
Example (read data in the database).....	13
Database Operations – Write/Save .....	14
OpenDataWriter.....	14
AddData.....	14
AddFormData .....	14
SaveData .....	15
Example (Add a record in the database).....	15
Example (Update a record in the database).....	16
Request Objects (QueryString, Form, Cookie) .....	17
getRequest.....	17
getForm.....	17
getCookie.....	17
Working with Full Source Code.....	18
Installation.....	18
IMPORTANT LICENSING REMINDERS:.....	18
OTHER NOTES: .....	18
Terms & Conditions .....	19

# Getting Started

This Guide is intended for advanced users of PDshop with HTML and ASP.NET (VB.NET) experience. The language and examples in this document are geared toward advanced users. In order to make use of this document, you should have some level of prior programming experience.

## *PDshop Namespaces*

In order to use any of these functions in your own project or .aspx pages, you must import the appropriate Namespace into your page.

PDshop Edition:	Namespace:
(original) .Net Edition	PDshopNet
.Net Standard Edition	PDshopNetStandard
.Net Advanced Edition	PDshopNetAdvanced
Version 8	PDshop8
Express Edition	PDshop9
Professional Edition	PDshop9
Developer Edition	PDshop10

Below are some examples of how you would import the PDshop namespace into your asp.net page.

### For Developer Edition, use this code:

```
<%@ import Namespace="PDshop10" %>
```

### For Express and Professional Editions, use this code:

```
<%@ import Namespace="PDshop9" %>
```

### For Version 8, use this code:

```
<%@ import Namespace="PDshop8" %>
```

## *PDshop Functions Class*

All functions discussed in this document are in the PDshop functions Class, the name of the class depends on your version you are using.

PDshop Edition:	Class Name:
(original) .Net Edition	PDshopNetFunctions
.Net Standard Edition	PDshopNetFunctions
.Net Advanced Edition	PDshopNetFunctions
Version 8 (and higher)	PDshopFunctions

Below are some examples of how you would bring the PDshop's functions into your page.

### For version 8 (and higher), your code would look like this:

```
Public pd As PDshopFunctions  
pd = New PDshopFunctions()
```

### For all older versions, your code would look like this:

```
Public pd As PDshopNetFunctions  
pd = New PDshopNetFunctions()
```

## Sample ASPX script

Below is an example of a typical Version 9 ".aspx" page that includes the PDshop9 namespace and PDshopFunctions class.

### Express & Professional Edition (Version 9) Example

```
<%@ Page Language="VB" Explicit="False" %>
<%@ import Namespace="PDshop9" %>

<script runat="server">

    Public pd As PDshopFunctions

    Sub Page_Load(Sender As Object, E As EventArgs)

        pd = New PDshopFunctions()
        pd.LoadPDshop()

        customerid=pd.getcookie("customerid")
        pd.getCustomerVariables(customerid)

    End Sub

    Sub Page_UnLoad(Sender As Object, E As EventArgs)

        If isnothing(pd) = False Then pd.UnloadPDshop()

    End Sub

</script>

<html>
<body> YOUR HTML GOES HERE </body>
</html>
```

### Other Versions

For older versions of PDshop, just change the namespace and functions class names (see the previous section in this guide). If you are using the Developer Edition, Version 10, some of the code above is found in your template.master file instead. See next section.

# Sample Master Page

The Developer Edition of PDshop now supports Master Pages. Below is an example of a typical Version 10 "template.master" page that includes the PDshop10 namespace and PDshopFunctions class.

## Developer Edition Master Page Example

```
<%@ Master Language="VB" Explicit="False" %>

<%@ Import Namespace="PDshop10" %>
<script runat="server">

    Public pd As New PDshopFunctions
    Public pgtype As String

    Sub Page_Error(ByVal sender As Object, ByVal e As System.EventArgs)

        pd.PDShopError()

    End Sub

    Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs)

        'Get Master Page variables
        pd.getMasterHtml(pgtype)

    End Sub

    Sub Page_UnLoad(ByVal Sender As Object, ByVal E As EventArgs)

        pd.UnLoadPDshop()

    End Sub

</script>
<html>
<body>
    YOUR HTML GOES HERE
    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
    </asp:ContentPlaceHolder>
    YOUR HTML GOES HERE
</body>
</html>
```

## Installation

To fully enable Master pages in PDshop, you first must add the following Key to your web.config file's appSettings section:

```
<add key="shopmaster" value="ON" />
```

## Other Versions

For older versions of PDshop that do not support Master Pages, see the .aspx page example in the previous section.

# Customer Functions

## *getCustomerVariables*

When called, this function will retrieve the customer's profile information (billing address, email, login, password, etc) and the data will become accessible thru some pre-defined public variables.

### Function Syntax

getCustomerVariables(customerid)

<u>Parameter</u>	<u>Example</u>	<u>Description/Notes</u>
iCustomerid	1	The unique "id" for this customer. The customerid corresponds to the "id" in the "customers" database table. Not required if calling order by cartid.

### Example

pd.getCustomerVariables(customerid)

The above example shows how you can call this function in a typical page. Before calling this function, you need to define the value of customerid. For example, if the customer is signed in already, you may want to first get the customerid from the web browser using the "getcookie" function, like this:  
customerid=pd.getcookie("customerid").

### Public Variables

Below is a list of all the variables that will become public after this function is called.

To access the variables in your .aspx page, you would use syntax similar to this:  
yourvariable = pd.sVariableName

Hints:

Replace "sVariableName" with the variable.

If a variable begins with "s", this denotes a String value.

If a variable begins with "i", this denotes an Integer.

<u>Variable</u>	<u>Example</u>	<u>Description/Notes</u>
iCustomerid	1	Corresponds to the "id" in the customer table
sEmail	demo@pdshop.com	Email address (used to login)
sPassword	abc123	Customer's Password
sPhone	555-555-5555	Customer's Phone
sName	John Smith	Billing/Profile Address: This address is common to all orders (appears as the billing address)
sFirstName	John	
sLastName	Smith	
sStreet1	555 Test Street	
sStreet2	Apt. 5	
sCity	New York	
sState	NY	
sCountry	United States	
sZip	10001	
sTempkey	123456789987654321	Unique key used to identify & authenticate user.

# Order Functions

## *getOrderVariables*

When called, this function will retrieve an order and calculate the order totals. Information related to the order is made accessible thru some pre-defined public variables.

### Function Syntax

getOrderVariables(customerid, orderid, cartid)

<u>Parameter</u>	<u>Example</u>	<u>Description/Notes</u>
iCustomerid	1	The unique "id" for this customer. The customerid corresponds to the "id" in the "customers" database table. Not required if calling order by cartid.
orderid	1	The unique "id" that identified this order. This "id" corresponds to the "id" column of the "orders" database table. When calling an order by this id, you must also provide the customerid (as a security measure). Not required if calling order by cartid.
cartid	"123456987654321"	This is the unique number (usually very long), that identifies the shopper's shopping cart. When you call by cartid, only very basic information about the shopping cart contents are retrieved (such the number of items, item weight, and item subtotal).

### Example

```
pd.getOrderVariables(customerid, orderid, cartid)
```

### Public Variables

Below is a list of all the variables that will become public after this function is called.

To access the variables in your .aspx page, you would use syntax similar to this:  
yourvariable = pd.sVariableName

#### Hints:

Replace "sVariableName" with the variable.

If a variable begins with "s", this denotes a String value.

If a variable begins with "i", this denotes an Integer (Int32) value.

If a variable begins with "d", this denotes a Decimal (Double) value.

<u>Variable</u>	<u>Example</u>	<u>Description/Notes</u>
iOrderid	1	Corresponds to the "id" in the orders table
iCustomerid	1	Corresponds to the "id" in the customer table
iOrderno	1	The Unique Order Number assigned by PDshop
sOrderdate	1/1/2007	Date of order
sStatus	New	Initial status of all orders in PDshop is "Incomplete". When checkout is completed, all orders are automatically moved to status of "New".
sShipName	John Smith	Shipping Address:
sShipFirstname	John	For billing address, see the getCustomerVariables
sShipLastname	Smith	function or see the customer database table.
sShipstreet1	555 Test Street	

sShipstreet2	Apt. 5	
sShipcity	New York	
sShipstate	NY	
sShipcountry	United States	
sShipzip	10001	
sCcnoc	John A. Smith	Name on credit card (as entered by user)
sCctype	Visa	Credit card type or Payment method (Pay by Mail, PayPal, etc)
sCcardno	4100000000000000	Full credit card number *
sCcardnohide	4100****0000	Protected credit card number (partially hidden) *
sExpmonth	01	Expiration Month
sExpyear	2007	Expiration Year
sCccode	123	3-4 Digit security code *
sCccodehide	***	Protected 3-4 Digit security code (hidden) *
sCcapproval	123456	Approval code captured by Authorize.net or PayPal Website Payments Pro
sCctrxid	1234567890123456789	Transaction ID captured by Authroize.net or PayPal
sIpaddr	10.10.1.100	The IP Address of the customer (as reported by your web server, captured at start of checkout)
iItemcount	1	The number of items in the order (does not include "options", parent items only)
dOrderweight	1.001	Total weight of all items in order
dOrdersub	1.01	Order subtotal, the sum of all price*quantity
dDiscamt	1.01	Flat discount amount to be applied to order
dDiscpct	1.001	Percentage discount to be applied to order
dDiscounts	1.01	Total amount of discount (Flat & Percent calculation)
dCarrierrate	1.01	Shipping Charges
dSalestax	1.001	Sales tax rate
dOrdertaxtotal	1.01	Total amount of tax charged (based on sales tax rate)
dOrdertotal	1.01	Order total before gift certificate has been applied
dGiftamt	1.01	Gift certificate amount (if part or all of order was paid with gift certificate)
dOrdergrandtotal	1.01	Grand Total (amount due or charged to credit card)
sNotes		Private Notes
sNotes2		Public Notes (notes for customer to see)
iShipresdel	1	Indicates whether shipping address is Residential or commercial address (1=Residential, 0=Commercial)
sCarriername	UPS Ground	Shipping Service/Carrier name
sTaxship	Y	Indicates whether the shipping charge is taxable
sTracking	1234567890123456789	Tracking Number (that you entered in PDAdmin)
sGiftcode	ABCDEFGH	Gift certificate code
sDisccode	ABCDEFGH	Discount code

\* Depending on your security settings, some data may not be available, such as credit card or card code numbers.



# Email Functions

## SendEmail

This function can be called to send an email. Before using this function, be sure you have setup and enabled Email in PAdmin (a valid SMTP server address must be provided in PAdmin)

### Function Syntax

pd.sendemail(emailto, emailcc, emailfrom, emailfromname, emailsubject, emailbody, emailformat, efilename)

<u>Parameter</u>	<u>Example</u>	<u>Description/Notes</u>
emailto	"person@yahoo.com"	The primary email address
emailcc	"otherperson@yahoo.com"	You can enter 1 "cc" email address
emailfrom	"demo@pdshop.com"	The "Sender" email address
emailfromname	"PDshop Demo"	The "Sender" name, can be left blank.
emailsubject	"RE: Your Order"	Subject of email
emailbody	(any text)	Body of Message
emailformat	"Text"	Email Format type, Text or HTML. To send email in HTML format you must (must) format the emailbody properly with all proper HTML tags and encoding; see W3C HTML guidelines.
efilename		This feature is not supported yet, future use.

### Example

```
pd.sendemail("person@yahoo.com", "2ndperson@yahoo.com", "demo@pdshop.com", "PDshop Demo", "RE: The Demo of PDshop", "Hello, the demo is great, I can send email!", "Text", "")
```

# Database Operations – Show data in your .aspx page

## *BindThisSQL*

This function can be used to bind (show) any data from the PDshop database, in your .aspx page. This function is extremely useful, and provides unlimited possibilities.

### Function Syntax

BindThisSQL(Sql)

<u>Parameter</u>	<u>Examples</u>	<u>Description/Notes</u>
Sql	"SELECT * FROM items"	Any valid SQL statement

### Example

The example below shows you how you might bind some items to your page using ASP.Net's built-in Repeater functions.

You might place code like this in the <script> section of your .aspx page:

```
mydata.DataSource = pd.bindthissql("SELECT * FROM items")
mydata.DataBind()
```

And place code like this in the HTML <body> section of your .aspx page:

```
<asp:Repeater id="mydata" runat="server">
  <itemtemplate>
    <%# DataBinder.Eval(Container.DataItem, "name") %><br>
    <%# DataBinder.Eval(Container.DataItem, "shortdesc") %>
  </itemtemplate>
</asp:Repeater>
```

# Database Operations – Run/Execute any SQL

## **SQLExecute**

This function can be used to run any SQL you wish on the PDshop database. This function is extremely useful, in that you can manipulate the data any way you wish.

### **Function Syntax**

SQLExecute(Sql)

<b><u>Parameter</u></b>	<b><u>Examples</u></b>	<b><u>Description/Notes</u></b>
Sql	"UPDATE customer SET name='John Smith' WHERE id=1"	Any valid SQL statement

### **Example**

If creating a new order record:

```
pd.SQLExecute("UPDATE customer SET name='John Smith' WHERE id=1")
```

# Database Operations - Read

## OpenDataReader

When called, this function will open the database and retrieve a specified row(s) of data for reading. This function is for use in conjunction with ReadData, and CloseData

### Function Syntax

OpenDataReader(Sql)

<u>Parameter</u>	<u>Example</u>	<u>Description/Notes</u>
Sql	"SELECT * FROM orders WHERE id=1"	Should consist of a valid SQL statement; typically a well formed "Select" statement.

### Example

```
pd.OpenDataReader("SELECT * FROM orders WHERE id=1")
```

## ReadData

This function will return the value of the database column for the row of data; you must call the OpenDataReader function before using this.

### Function Syntax

ReadData(column)

<u>Parameter</u>	<u>Example</u>	<u>Description/Notes</u>
Column	"shipname"	A database column (field) name

### Example

```
Myvar = pd.ReadData("shipname")
```

In this example, your variable is populated with the customer's name.

## ReadData (other variations)

There are several variations of the ReadData function, each designed for a different datatype.

<u>Function</u>	<u>Example</u>	<u>Description/Notes</u>
ReadData	Pd.ReadData(column)	Returns the data as a Text/String
ReadDataN	Pd.ReadDataN(column)	Returns the data as an Integer/Int32
ReadDataD	Pd.ReadDataD(column)	Returns the data as a Decimal/Double (formatted according to your number settings)
ReadDataC	Pd.ReadDataC(column)	Returns the data as a Decimal/Double (formatted according to your currency settings)

## CloseData

This function is called to close the DataReader. You must close the current datareader before performing any subsequent database operations.

### Function Syntax

CloseData()

### Example

```
pd.CloseData()
```

### ***Example (read data in the database)***

Below is an example of how you might use the OpenDataReader, ReadData, and CloseData functions together.

### Example

In the example below, we open the orders table, and read the shipping address' name.

```
pd.OpenDataReader("SELECT * FROM orders WHERE id=" & orderid & " AND customerid=" & customerid)
  If pd.ReadDataItem.Read Then
    Name.text = pd.ReadData("shipname")
  End If
pd.CloseData()
```

# Database Operations – Write/Save

## OpenDataWriter

When called, this function will open a specified database table so that you can add a new row, or update an existing row/record of data.

### Function Syntax

OpenDataWriter(Sqlpart)

<u>Parameter</u>	<u>Examples</u>	<u>Description/Notes</u>
Sqlpart	"orders" OR "orders WHERE id=1"	Should consist of the database table name (alone) if you are adding a new record (row of data), or the table name followed by a proper WHERE clause if updating an existing record.

### Example

If creating a new order record:  
pd.OpenDataWriter("orders")

If updating an existing order:  
pd.OpenDataWriter("orders WHERE id=1")

## AddData

This function will return let you add data that you will be saving; you must call the OpenDataWriter function before using this.

### Function Syntax

AddData(column, data, datatype)

<u>Parameter</u>	<u>Example</u>	<u>Description/Notes</u>
Column	"shipname"	A database column (field) name
Data	"John Smith"	The data you are saving
Datatype	"T"	This determines what type of data it is Text, Number, or Currency.  T=Text/String N=Numeric (Double/Integer) C=Currency X=Text/String (will be encrypted)

### Example

pd.AddData("shipname", "John Smith", "T")  
pd.AddData("carrirate", 5.00, "N")

## AddFormData

This function is almost the same as the AddData function, however it will check the data (assuming it is being submitted from a form), and report any data entry errors; you must call the OpenDataWriter function before using this. This function will report data entry errors in the public variable "pd.formerror". For example, if the data entered does not meet the minimum required characters, PDshop will report a message similar to this "[columnname] is incomplete or too short!"

## Function Syntax

AddFormData(column, data, columnname, datatype, datamin, datamax)

<u>Parameter</u>	<u>Example</u>	<u>Description/Notes</u>
Column	"shipname"	A database column (field) name
Data	"John Smith"	The data you are saving
Columnname	"Name"	The casual name, description, for the form field. This text is used in any messages during the form validation.
Datatype	"T"	This determines what type of data it is Text, Number, or Currency.  T=Text/String N=Numeric (Double/Integer) C=Currency E=Email address X=Text/String (will be encrypted)
Datamin	1	The minimum number of required characters (only applies when datatype is "T").
Datamax	10	The maximum number of characters allowed (only applies when datatype is "T").

### Example

```
pd.AddFormData("shipname", "John Smith", "Name", "T", 5, 25)
pd.AddFormData("carrierrate", 5.00, "Shipping Charge", "N", 0, 0)
```

## **SaveData**

This function is called to save & close the data you just added using the AddData and AddFormData functions. This function returns the "id" of the record just added when adding a new record.

### Function Syntax

SaveData()

### Example

```
pd.SaveData()
OR
Myid = pd.SaveData()
```

## **Example (Add a record in the database)**

Below is an example of how you might add a record to the database.

### Example

In this example we are adding a new record to the "customer" table in the database. The data is added from a form that is submitted. You can use the AddData function rather than the AddFormData if you do not want to have the data validated.

```
pd.OpenDataWriter("customer")
pd.AddFormData("name", name.Text, "Name", "T", 3, 50)
pd.AddFormData("email", email.Text, "Email Address", "E", 5, 50)
```

[repeat above example for each column of data]

```
if pd.formerror="" then
```

```
customerid = pd.SaveData()  
end if
```

### ***Example (Update a record in the database)***

Below is an example of how you might add a record to the database.

#### **Example**

In this example we are updating a record in the "customer" table in the database. The data is added from a form that is submitted. You can use the AddData function rather than the AddFormData if you do not want to have the data validated.

```
pd.OpenDataWriter("customer WHERE id=1")  
pd.AddFormData("name", name.Text, "Name", "T", 3, 50)  
pd.AddFormData("email", email.Text, "Email Address", "E", 5, 50)
```

[repeat above example for each column of data]

```
pd.SaveData()
```



# Request Objects (Querystring, Form, Cookie)

## *getRequest*

This function is called to capture Querystring or Form data that has been posted/sent to the page.

### Function Syntax

```
getRequest(variablename)
```

<u>Parameter</u>	<u>Examples</u>	<u>Description/Notes</u>
Variablename	"name"	

### Example

```
Myvar = pd.getRequest("name")
```

## *getForm*

This function is called to capture Form data that has been posted to the page.

### Function Syntax

```
getForm(variablename)
```

<u>Parameter</u>	<u>Examples</u>	<u>Description/Notes</u>
Variablename	"name"	

### Example

```
Myvar = pd.getForm("name")
```

## *getCookie*

This function is called to capture cookie data that has been previously saved to the user's web browser.

### Function Syntax

```
getCookie(variablename)
```

<u>Parameter</u>	<u>Examples</u>	<u>Description/Notes</u>
Variablename	"something"	

### Example

```
Myvar = pd.getCookie("something")
```

## Working with Full Source Code

### *Installation*

When you purchase the additional Source Code for PDshop, you will receive a special .zip file. After you receive this file, there are only a few simple steps you will need to follow in order to use the PDshop source code.

1. Save and Extract the contents of the Source Code zip file to a safe location on your computer.
2. Locate the "App\_Code" folder you just extracted, and upload this folder to your web site's web root, for example "your\*web\*address.com/App\_Code"
3. In order for the PDshop pages to recognize and start using the source code, you must modify the Shop and Admin "template.master" files on your server. You will simply need to change the PDshop Namespace that is being imported.

Locate this line of code:

```
<%@ Import Namespace="PDshop10" %>
```

And change it to:

```
<%@ Import Namespace="PDshop10_Source" %>
```

### ***IMPORTANT LICENSING REMINDERS:***

Under no circumstances are you allowed to share or distribute any copies of this source code. The source code is provided to you, for your use only. This code is intended for use with your licensed copies of PDshop only.

### ***OTHER NOTES:***

If your copy of PDshop is installed in a subfolder (and that subfolder has been configured as an application directory), then you may need to place the "App\_Code" folder into that directory instead.

Do not modify or remove the original "bin" folder. The "PDshop10.dll" file must remain on your server at all times in order for PDshop to work.

## Terms & Conditions

This guide is intended for licensed users of PDshop only.

See the Terms & Conditions for PDshop on the PageDown Technology website, see <http://www.pagedowntech.com/terms>.

Additional Support and Information about this product can be found on PageDown Technology's website, <http://www.pagedowntech.com>.

### Copyright Notice:

All Text and Images contained on this document is the property of PageDown Technology, LLC ([pagedowntech.com](http://pagedowntech.com)) and cannot be reproduced, copied, or used without our written permission.

PageDown Technology, LLC / Copyright 2003-2012 All Rights Reserved