

Creating an Order History page for your customers

Order History pages can be very useful, they allow your customers to keep track of what they have purchased, and can be further enhanced to function as a Download Center if you are selling digital goods.

This How To guides you through creating an Order History page, and a customer's Order Detail page to display their orders.

Note: In order to successfully accomplish the steps describe in this document, you should first follow the steps in the following How To in order to implement register and login functionality for customers.

http://www.webassist.com/go/howto/ecart_secass

What Do You Need to Start?

- Dreamweaver 8, CS3, or CS4
- eCart
- DataAssist
- SecurityAssist

Record Customer ID in Database

The first step is to ensure that the Customer ID for the customer who is making the purchase, is going to be recorded in the database along with the details for the order.

Note: You will need to have configured eCart to store the cart summary in your database. You can confirm this by checking to see if an *eCart Store Summary in Database* server behavior exists in the Server Behaviors panel of the confirm page.

Creating the CustomerID column in your database

First, you need to create the database column in your orders table for storing the customer/user ID.

1. Using your preferred database manipulation tool, open the **orders** table that you are using to store your order summary.
2. Insert a new column to your database, name this column **OrderUserID**.
3. Specify this column as the **Integer** Datatype.

Update the server behavior

4. In Dreamweaver, open your eCart **confirm** page.
5. From the Server Behaviors panel, double-click on the **eCart Store Summary in Database** server behavior.
6. Select the **OrderUserID** column.
7. Click the Value lightning bolt.
8. Expand the **Session** option.
9. Select the **UserID** option.
Depending on the Session Variable used with SecurityAssist, this may be named differently.
10. Click **OK**.
11. Click **OK** again.

You've now configured eCart to store the customer ID in the database when it records each order. You will now use this information to create the Order History page.

Creating the Order History and Order Detail pages

In this next step, you will learn how to create the Order History page using DataAssist.

Creating the Order History page

1. Open the DataAssist wizard by choosing **Insert > WebAssist > DataAssist > DataAssist Wizard**.
2. Choose your database connection.
3. Choose your **orders** table
4. If desired, choose a template to use.
5. Choose to create a **Results** and **Detail** page
For this example, you do not need to create any of the other pages and should uncheck those options.
6. Name the results page **customer_order_history.php**.
Throughout this tutorial, we will use this filename for the results page. Feel free to choose a different filename.

7. Leave the Detail page with the default name.
This page will only be used as a placeholder for a short period of time.
8. Click **Next** to proceed through the wizard.
9. Proceed through the wizard as you would normally, removing any unnecessary database columns from being included on your page, and renaming any others.

Customizing the Order History page options

The following few adjustments should be made to the Results page options as you proceed through the wizard.

10. Make sure to choose *Link to detail page* for at least one of the database columns.
Typically, this would be the Order ID column.
11. From the Filter menu, choose **OrderUserID**.
This is the database column that stores the UserID for the order. *Your database column may be named differently.*
12. Choose the **Session Variable** option from the menu below.
13. Enter **UserID** in the text field.
In this example, UserID is the name of the Session Variable used by SecurityAssist.

Filtering and Sorting:			
Filter:	OrderUserID	Numeric	=
	Session Variable	UserID	
Sort:	None	Ascending	


14. Proceed through the rest of the wizard and click Finish to create the two pages.

Creating the Order Details Results page

Next, we will run through the DataAssist wizard again to create the results page which we will use to display the details for each order.

15. Open the DataAssist wizard by choosing **Insert > WebAssist > DataAssist > DataAssist Wizard**.
16. Choose your database connection.

17. Choose your **order details** table
18. This time, only choose to create a **Results** page.
Make sure to uncheck all other pages.
19. Name the results page **customer_order_details.php**.
20. Click **Next** to proceed through the wizard.
21. Proceed through the wizard as you would normally, removing any unnecessary database columns from being included on your page, and renaming any others.
22. Click **Finish** to create the page.

For more details on using the DataAssist Wizard, review the [DataAssist Getting Started Guide](#) .

Applying the SecurityAssist Access Pages Rule

In order to protect the DataAssist pages you just created, we need to apply SecurityAssist.

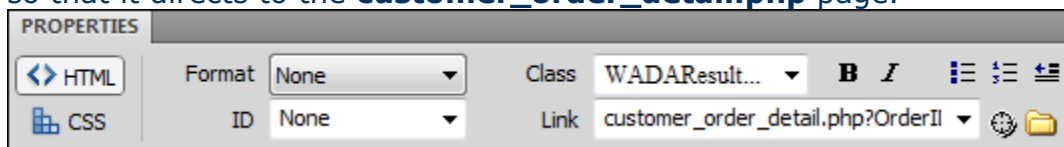
Before proceeding, make sure to close the pages DataAssist has created in Dreamweaver.

1. From the Modify menu, choose **SecurityAssist > Access Pages Manager**.
2. **Ctrl (Cmd) + Click** to select both the customer_order_detail.php and customer_order_history.php pages.
3. From the Grant access if menu, choose **Logged in to users**.
4. Choose the browse icon for *If access denied, go to*.
5. Select your **Login** page.
6. Click **OK**.
7. Click **OK** again to close the Access Pages Manager.

Customizing the Order History page

Next, we need to modify the Order History page to go to our Order Detail page.

1. Open the **customer_order_history.php** page.
2. In Design View, click once on the link that goes to the Detail page.
You specified this in DataAssist. Typically this would be the OrderID link.
3. In the Properties Inspector, modify the filename that the link goes to, so that it directs to the **customer_order_detail.php** page.



Note: You only need to adjust the filename at the beginning of the Link. The rest of the URL should not be touched.

4. Press **Ctrl (Cmd) + S** to save the page.

Optional: Now that this has been completed, you can delete the Detail page that you created the first time you ran through the DataAssist Wizard.

Customizing the Order Detail Results page

We will now customize the recordset used by the order detail page to connect both the orders and the order details tables. This will allow us to display the correct details for each specific order.

1. Open the **customer_order_detail.php** page.
2. From the Server Behaviors panel, double-click on the Recordset.

In the Recordset window, we will make a number of modifications. The following section is broken down into multiple steps for greater clarity.

Note: For each mention of a database table or column, you may need to adjust the new code to match your database's configuration.

The SELECT statement

The SELECT statement should currently appear as follows:

```
SELECT DetailID, DetailOrderID, DetailProductID,  
DetailName, DetailPrice, DetailSKU, DetailQuantity
```

Note: Depending on the options you selected in DataAssist, you may have different database columns listed.

1. Change this entire statement to read as follows:

```
SELECT orders.*, orderdetails.*
```

Summary:

```
orders.*
```

Display all database columns from the orders table.

```
orderdetails.*
```

Display all database columns from the order details table.

The FROM statement

The FROM statement should currently appear as follows:

```
FROM orderdetails
```

1. Change the entire statement to read as follows:

```
FROM orders INNER JOIN orderdetails ON orders.OrderID  
= orderdetails.DetailOrderID
```

Summary:

```
orders INNER JOIN orderdetails
```

This INNER JOIN statement connects both the orders and the order details tables.

```
ON orders.OrderID = orderdetails.DetailOrderID
```

This connects the OrderID column from the orders table with the DetailOrderID from the Order Details table.



For a greater understanding of how INNER JOIN statements work, review the following tutorial

http://w3schools.com/sql/sql_join_inner.asp.

The WHERE statement

The WHERE statement should currently appear as follows:

```
WHERE DetailID = ParamDetailID OR ( -1= ParamDetailID2  
AND DetailID= ParamSessionDetailID)
```

2. Change the entire statement to read as follows:

```
WHERE orderdetails.DetailOrderID = paramOrderID AND  
orders.OrderUserID = paramUserID
```

Summary:

```
orderdetails.DetailOrderID = paramOrderID
```

This connects the DetailOrderID column of the Order Details table with the parameter (paramOrderID) which we will configure in the next step.

```
AND orders.OrderUserID = paramUserID
```

This connects the OrderUserID column of the Order table with the parameter (paramUserID) which we will configure in the next step.

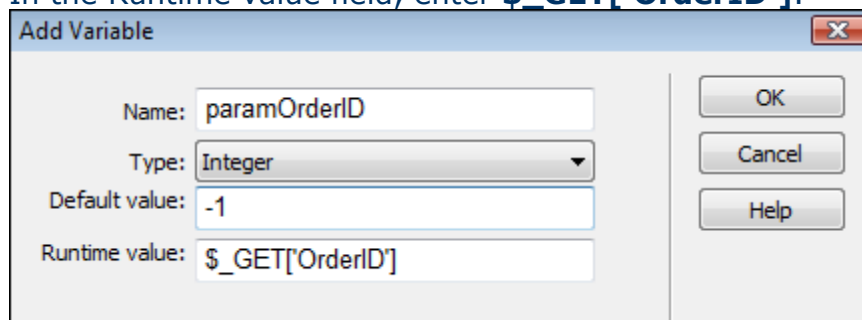
Once you have modified the entire SQL statement, it should appear like this:

```
SQL: SELECT orders.*, orderdetails.*  
FROM orders INNER JOIN orderdetails ON orders.OrderID =  
orderdetails.DetailOrderID  
WHERE orderdetails.DetailOrderID = paramOrderID AND orders.OrderUserID =  
paramUserID
```

Modifying the Variables (Parameters)

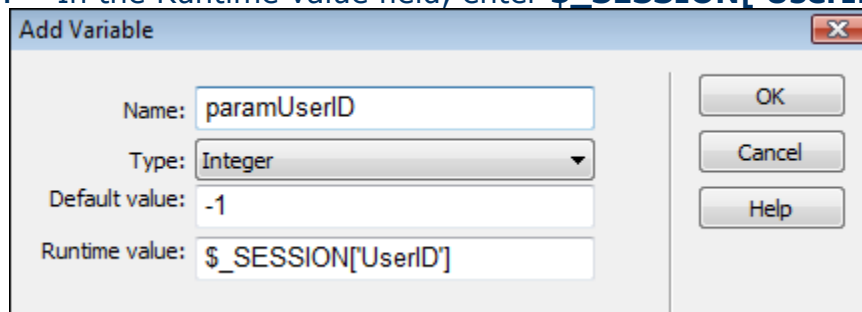
The final step in modifying the SQL statement is to modify the Variables being used by this recordset to work with our new SQL statement.

1. Click once to select the first variable, **ParamDetailID**.
2. Choose the **Delete (-)** button to remove this variable.
3. Repeat this process for the other two parameters, leaving the variables window empty.
4. Click **Add (+)** to create a new variable.
5. In the name field enter **paramOrderID**.
6. In the Default value field, enter **-1**.
7. In the Runtime value field, enter **\$_GET['OrderID']**.



The screenshot shows a dialog box titled "Add Variable" with a close button (X) in the top right corner. It contains four input fields: "Name:" with the text "paramOrderID", "Type:" with a dropdown menu showing "Integer", "Default value:" with the text "-1", and "Runtime value:" with the text "\$_GET['OrderID']". To the right of these fields are three buttons: "OK", "Cancel", and "Help".

8. Click **OK**.
9. Click **Add (+)** to create the next variable.
10. In the name field enter **paramUserID**.
11. In the Default value field, enter **-1**.
12. In the Runtime value field, enter **\$_SESSION['UserID']**.



The screenshot shows a dialog box titled "Add Variable" with a close button (X) in the top right corner. It contains four input fields: "Name:" with the text "paramUserID", "Type:" with a dropdown menu showing "Integer", "Default value:" with the text "-1", and "Runtime value:" with the text "\$_SESSION['UserID']". To the right of these fields are three buttons: "OK", "Cancel", and "Help".

13. Click **OK**.

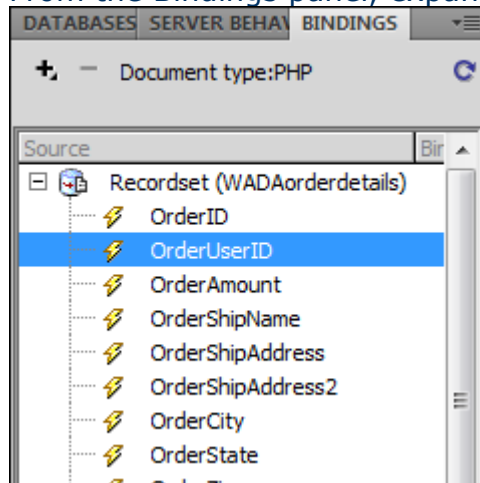
14. Click **OK** again to update your Recordset.
15. Press **Ctrl (Cmd) + S** to save your page.

Cleaning up the Order Details Results page

Now that you have updated your recordset, you will be able to display details on the Order Detail page from both the Orders table and the Order Details table.

Adding additional order details

1. From the Bindings panel, expand the WADAorderdetails Recordset.



2. Click and drag any of the Bindings to your page that you want to be displayed on the Order Details page.

Additional modifications

You may also want to remove certain details from displaying on the results page. For example, in most cases, displaying the SKU will not be necessary for most pages.

You've now completed the creation of an Order History page for your customers.

Note: For testing your Order History page, you will need to have orders in the database. Perform some test transactions on your site if necessary.