Getting Started With MX Kart

Table of Contents

Overview

In the **Getting Started with MX Kart** tutorial, you will learn how to use MX Kart for quickly creating an e-commerce site. Beginning with the files in your installation package, you will understand what MX Kart is, how it works, and what you can do with it.

Further sections of this tutorial contain instructions for using some of MX Kart's most popular features in a practical setting.

Although much of the tutorial and sample site were done in PHP, the ColdFusion steps are nearly identical. The primary difference is in the database configuration and connection sections, where separate instructions are provided.

About this Software

MX Kart provides a collection of commands and server behaviors that will create the shopping cart code for you. Web programmers will benefit from using our Kart Recordset functionality, because it is a native recordset that allows you to reuse the server behaviors already installed. Unlike other shopping cart systems, MX Kart is easy to use and extensible. It allows you to customize the e-commerce application logic directly from Dreamweaver MX or MX 2004.

MX Kart supports PHP_ADODB, PHP_MySQL, and ColdFusion server models. To keep up with the latest MX Kart information, visit:

http://www.interaktonline.com/Products/Dreamweaver-Extensions/MXKart/Overview/

What's Inside?

Folder Structure

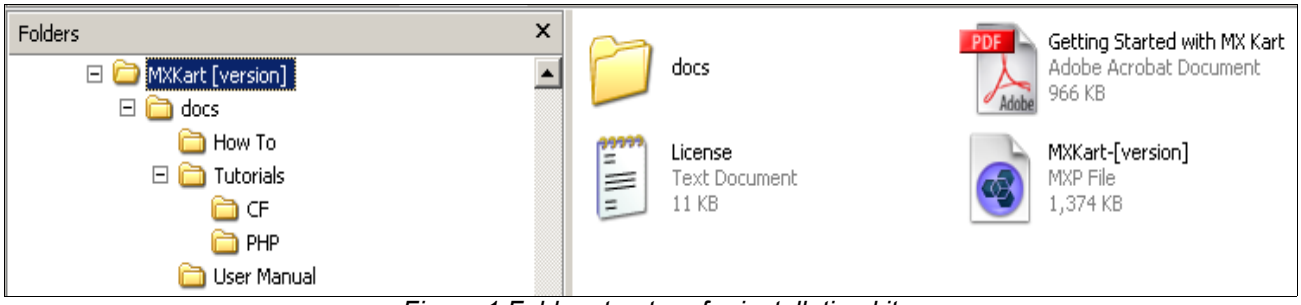


Figure 1 Folder structure for installation kit

The installation kit includes a root folder and one sub folder. The root folder is where you will find the MXKart [version] .mxp file for installation. Also in the root folder is the Getting Started with MX Kart tutorial, and a copy of the ownership license. Here is what the sub folders contain:

docs

This folder contains three sub folders with a great deal of information on MX Kart and how to use it. The folders and their contents are listed below:

How to

The How to folder contains a PDF file with instructions for configuring a payment gateway. The steps included are essential in establishing a working e-commerce site.

Tutorials

The Tutorials folder contains two versions of the MX Kart Tutorial. One for PHP and one for ColdFusion. Each of the tutorial sub folders contains a pdf tutorial, an sql folder with files for creating and populating database tables, and a sites folder with sample website for each supported server model.

User Manual

The User Manual folder contains a PDF MX Kart User Manual with many details about MX Kart and how it works. This file is the best source for technical information.

Installing MX Kart

The installation procedure for MX Kart is very simple.

1. In your installation kit, first open the MX Kart folder.
2. Once inside the folder, you will find a **MXKart-[version].mxp** file.
3. Double click on this file. The Macromedia Extension Manager will automatically begin installing the extension.

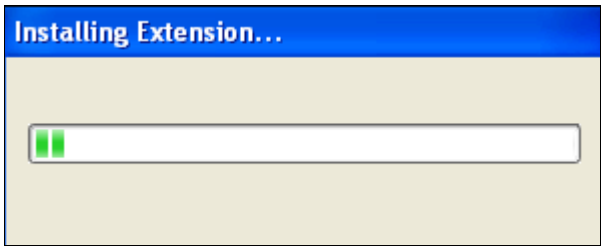


Figure 2 Extension Manager Installation Progress

4. A Macromedia disclaimer will appear. If you agree to the terms click the “Accept” button.

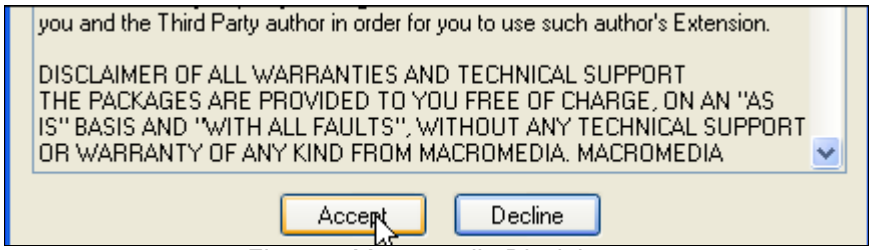


Figure 4 Macromedia Disclaimer

5. When the Extension Manager finishes installing (usually just a few seconds), a dialog box will appear notifying you that the installation is complete.

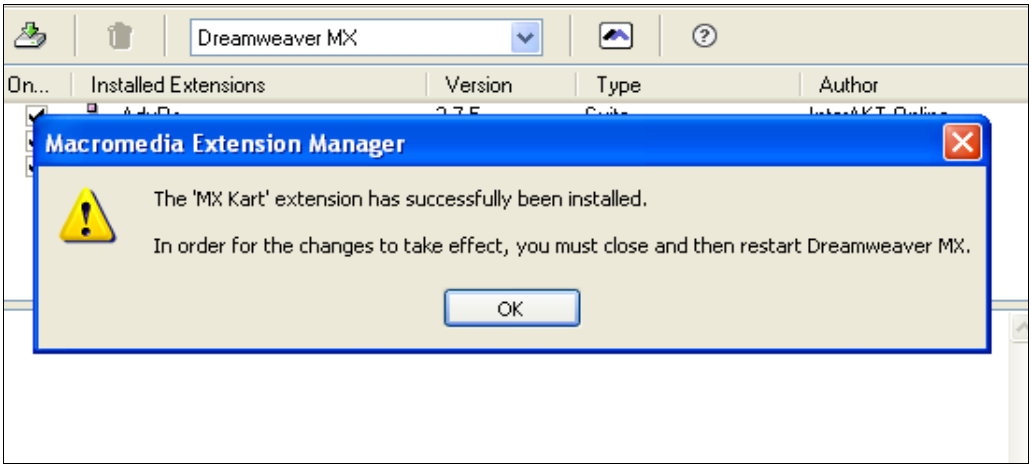


Figure 3 Installation Success

6. If you have Macromedia Dreamweaver open, you must close it then reopen it for the MX Kart elements to appear.

Uninstalling MX Kart

To uninstall **MX Kart** you must :

1. Open the **Macromedia Extensions Manager**.
2. In the Extensions Manager, select **MX Kart**.

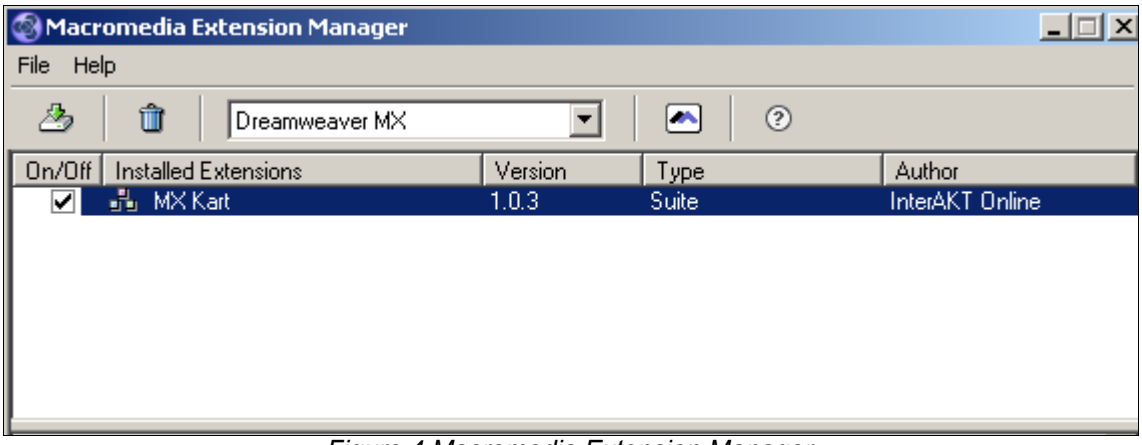


Figure 4 Macromedia Extension Manager

3. Click the **Remove Extension** icon (garbage can).

Figure 5 Remove extension icon

4. After selecting to remove the extension, a dialog box appears asking you to confirm the removal. Click **Yes.**

Figure 6 Confirmation window

5. Once you confirm, a window will appear displaying the removal progress.

Figure 7 Removal progress

6. When the Extension Manager finishes uninstalling, a dialog box will appear to notify you that the removal is complete.

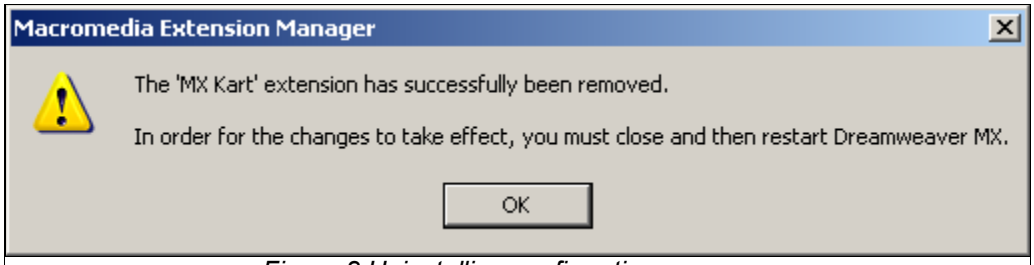


Figure 8 Uninstalling confirmation message

7. If you have Macromedia Dreamweaver open, you must close it then reopen it for the MX Kart elements to be completely removed.

Upgrading MX Kart

To upgrade MX Kart, complete the uninstalling procedure from the previous chapter. Once the uninstall finishes, install the new .mxp file using the instructions in **Chapter 4.**

In order to do a correct upgrade of MX Kart you should use the following steps for each site created using MX Kart:

1.Uninstall AdvRs. The current MX Kart release has AdvRS integrated in the installation file.

2.Uninstall the old MX Kart version using the Macromedia Extension Manager.

3.Install the new version by executing the .mxp file in your release folder.

4.In Dreamweaver, open an existing page containing MX Kart elements. You will be prompted for an update. Choose “yes” and the site will be updated.

Enjoy the updated version.

Exploring MX Kart

In this section, you will begin to explore the MX Kart workspace, and learn how to use its most common functionalities. Also included are steps for configuring your site and database connection so that you can start using MX Kart right away.

Workspace Orientation

There are two areas where MX Kart elements appear. The first is in the Insert bar. At the far right you will see a tab labeled MX Kommerce. When the tab is clicked, two MX Kart buttons appear. Later in this tutorial you will use the "Create shopping cart View" button.

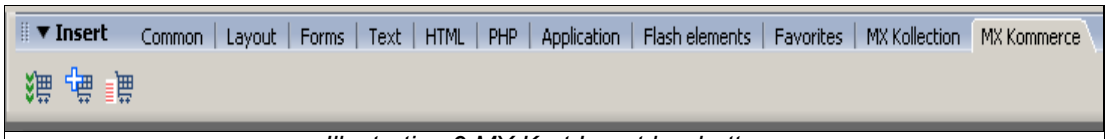


Illustration 9 MX Kart Insert bar buttons

The second area to find MX Kart elements is in the Server Behaviors tab of the Application panel.

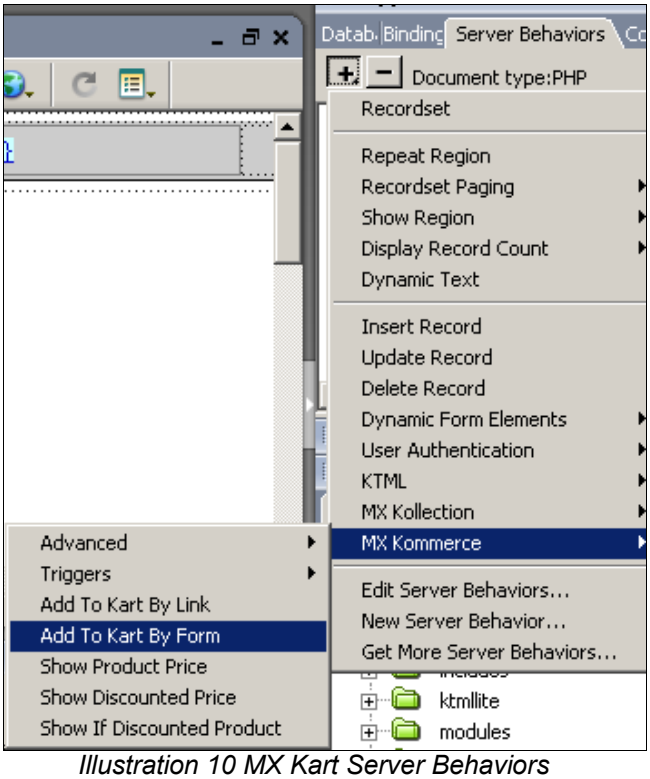


Illustration 10 MX Kart Server Behaviors

To start creating your site using MX Kart elements, there are three things that must be done first:

1. setting up your database
2. configuring the site
3. creating a database connection.

The first part, setting the database can be done by all users, while the instructions for the next two parts will depend on your server model.

If you already have a database, site, and connection set up, feel free to skip this section and move on to the

Creating a Product List, or **Creating a Shopping Cart** sections. Keep in mind that you will need to substitute your own variable names where appropriate, to match your server configuration.

Application Scenario

In this tutorial, you will build the product list and the shopping cart sections of an e-commerce site. Users will be able to view the products, add products to their cart, and view the cart contents with options to remove or add items. You will learn how easy it is to create these functionalities with MX Kart:

- Build a product list
- Add items to cart
- View cart contents
- Modify cart contents (quantities and items)

Setting up the Database Tables

For this tutorial, you'll use a database named **`mx_shop`**, containing 24 tables. While the database may seem complicated, you will only use a small subset of tables for this tutorial. The remaining tables will be used in the more detailed **MX Kart Tutorial** where you will create a complete e-commerce site handling discounts, multiple currencies, tax rates, shipping methods, and more.

The complete scripts used to generate the database tables are included in the MX Kart distribution package.

Please find below the graphical representation of the database table made using MX Query Builder:

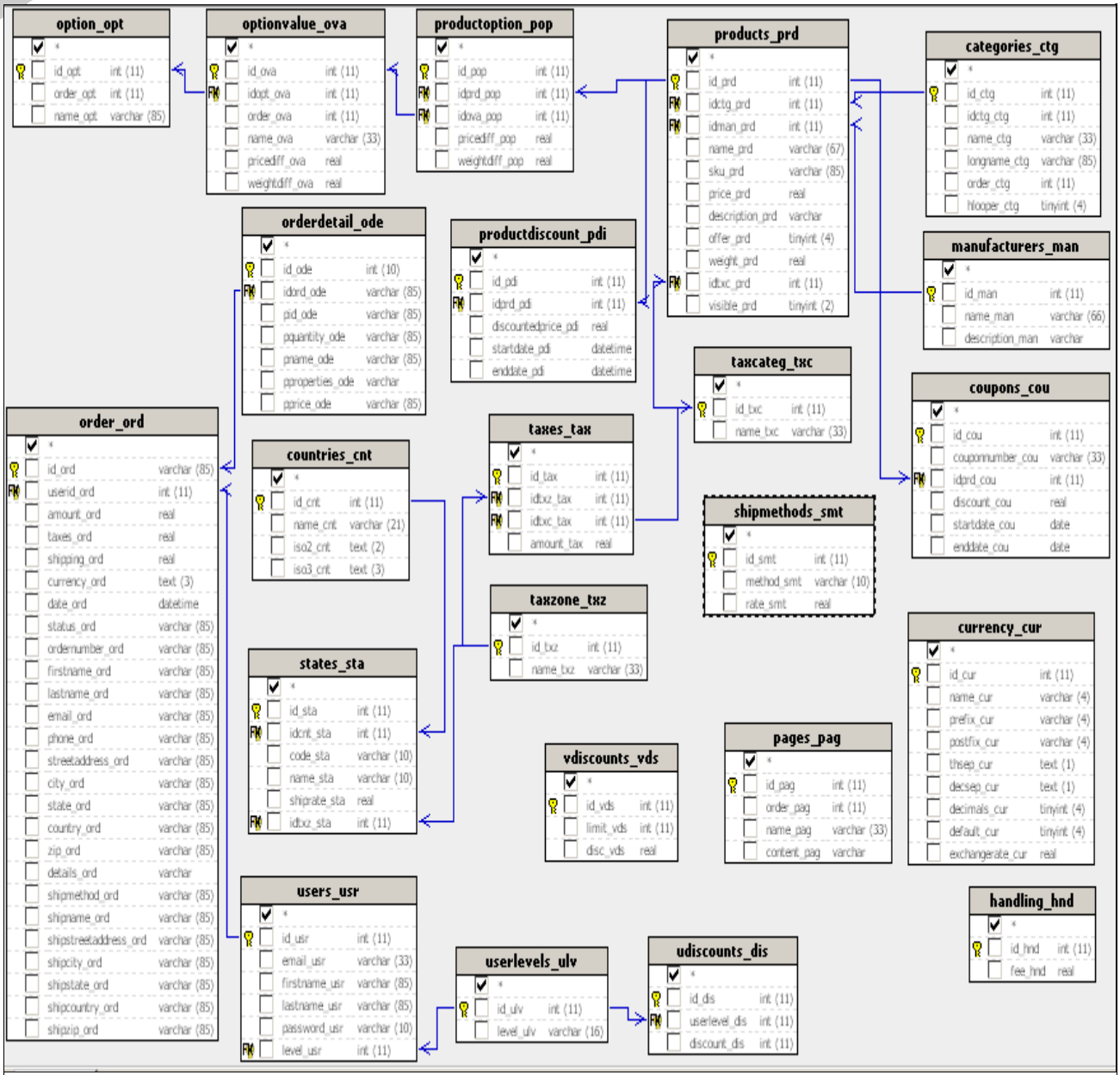


*Figure 11 The mx_shop Database **QuB** Representation*

Tips

MX Query Builder is an InterAKT Online product which provides a visual interface for generating complex SQL queries and managing table relations. Check out http://www.interaktonline.com/products/QuB for more information.

For this tutorial you will only be using two tables: *products_prd, and categories_ctg.* Here is a description of the tables:

- *products_prd* – this table stores all products to be sold on the website. The table fields are: ***id_prd*** (the primary key), ***idctg_prd*** (the foreign key to the *categories_ctg* table), ***idman_prd*** (the foreign key to the *manufacturers_man* table), ***name_prd*** (the product name), ***sku_prd*** (the product code), ***price_prd*** (the product price), ***description_prd*** (the product long description), ***offer_prd*** (if this field value is set to "1", the product is a special offer), ***weight_prd*** (the weight of the current product – will be used to compute the shipping costs), ***idtxc_prd*** (a foreign key to the tax categories table to select the tax category of the current product) and ***visible_prd*** (if this field value is set to "1", the product is visible on the site and can be bought by the customers

- *categories_ctg* – this table stores all product categories. The table fields are: ***id_ctg*** (the primary

key), ***idctg_ctg*** (parent category ID), ***name_ctg*** (the category name), ***longname_ctg*** (the category long name) ***order_ctg*** (the category order in the current subcategory) and ***hlooper_ctg*** (when 1, this can instruct the product list for a category to be rendered in a horizontal looper approach).

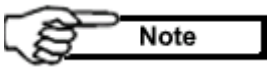
The remaining tables will be used in the more detailed **MX Kart Tutorial.**

Site Configuration

Open Macromedia Dreamweaver MX. Using the "New Site" option from the "Site" menu, create a new site named "mx_kart" Configure your site following the next three steps.

Configuring the Local Info section:

The Local Information section is where you define the location where you store all site files before uploading them to the server.



For the site configuration section, you must provide your own settings. The following list is just an example.

- **Site Name**: mx_kart
- **Local Root Folder**: C:\www\mx_kart\
- **Refresh Local File List Automatically**: Checked
- **Default Images Folder**: You can either enter the name of your images folder, or leave this blank
- **HTTP address**: http://my_server/mx_kart/
- **Cache**: enabled

Configuring the Remote Info section:

- **Access**: Local/Network or FTP
- **Remote Folder**: Path to the remote server. If using FTP, enter the connection details
- **Refresh Remote File List Automatically**: checked
- **Automatically upload files to the server on save**: checked

Configuring the Testing Server section:

- **Server Model**: You must provide your own server model. In this tutorial, **PHP_ADODB** was used.
- **Access**: Local/Network
- **Testing Server Folder**:V:\test\mba\mx_breadcrumbs\
- **Refresh Remote File List Automatically**: Checked
- **URL Prefix**:http://work.iakt.ro/test/mba/mx_breadcrumbs/

Database Connection

Before setting up the database connection, you should create the pages used in this tutorial:

- `index.php/index.cfm`

- *prodDetail.php/prodDetail.cfm*
- *viewKart.php/viewKart.cfm*

Content for these files will be completed later in the tutorial.

To set up your database connection, select the appropriate section according to your server model. The other two may be skipped. Sections 4.5.1, 4.5.2, and 4.5.3 contain instructions for PHP_ADOdb, PHP_MySQL, and ColdFusion respectively.

PHP_ADOdb

Create a new database connection by using the **ADODB Connection** option, that can be found using the Plus (+) button from the **Databases** tab in the **Application** panel. This option is available only to PHAkt2 users.

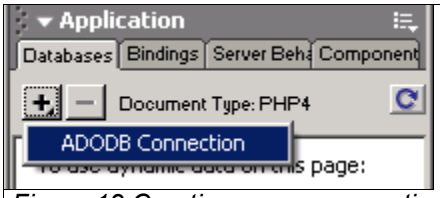


Figure 12 Creating a new connection

Then follow the example below to properly configure the connection:

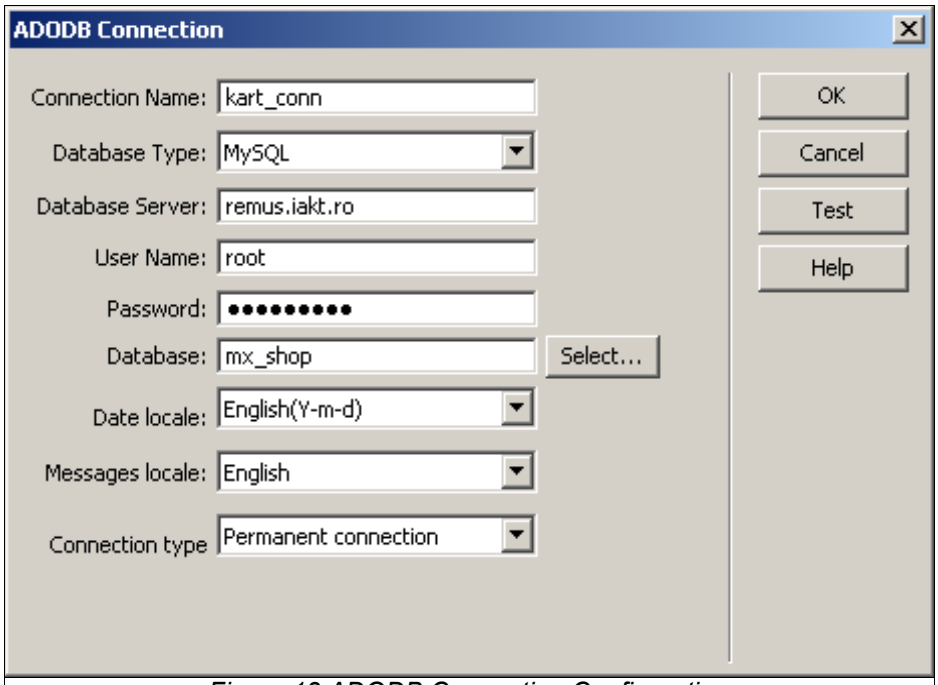


Figure 13 ADODB Connection Configuration

PHP_MySQL

If you use PHP and MySQL, create a MySQL connection:

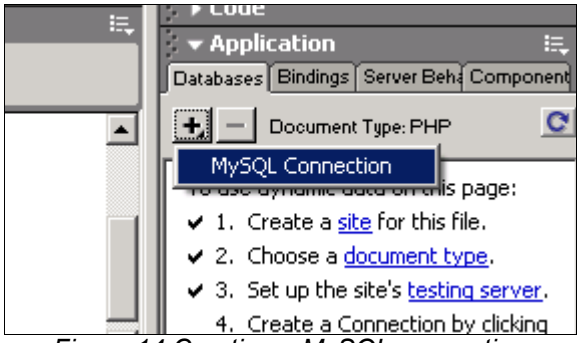


Figure 14 Creating a MySQL connection

Follow the below example to properly configure the MySQL connection:

Please replace the ***Database Server***, ***User Name,*** and ***Password*** field values to reflect your configuration settings.

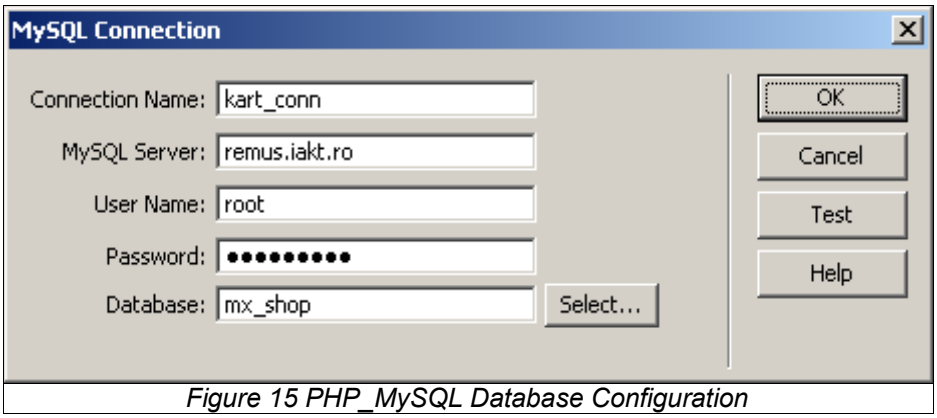


Figure 15 PHP_MySQL Database Configuration

ColdFusion

In order to be able to query the **`mx_shop`** database to retrieve information, you have to create a system DSN on the testing and/or remote server. You will use the mx_shop.mdb script provided with this tutorial. The Data Source Name (DSN) provides connectivity to a database through an ODBC driver. The DSN contains database name, directory, database driver, user ID, password, and other information. Once you create a DSN for a particular database, you can use the DSN in an application to call information from the database.

To create a system DSN on the ColdFusion server:

1. In Windows, from the **Start** menu go to **Settings->Control Panel->Administrative Tools** and double-click **Data Sources (ODBC).**
2. Select the ***System DSN*** tab. Click on the **Add** button.
3. In the displayed window, select ***Microsoft Access Driver (*.mdb)*** as driver for which you want to set up a data source and click **Finish**.
4. In the configuration window that is displayed:
    - Enter mx_shop in the ***Name*** field
    - Navigate to the **mx_shop.mdb** database script provided with this tutorial, using the **Select** button, as shown below:

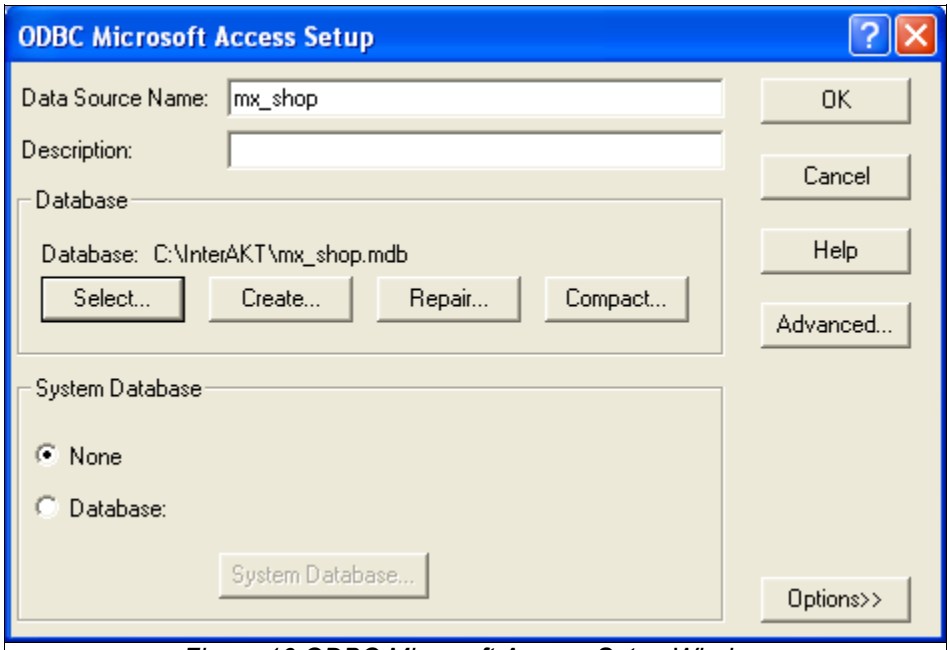


Figure 16 ODBC Microsoft Access Setup Window

5. Click **OK** when done, than click **OK** once again.

Next, after you open the `index.cfm` page, you will notice in the **Bindings** tab of the **Application** panel a checklist with the settings that are to be made in order to use dynamic data on the page:

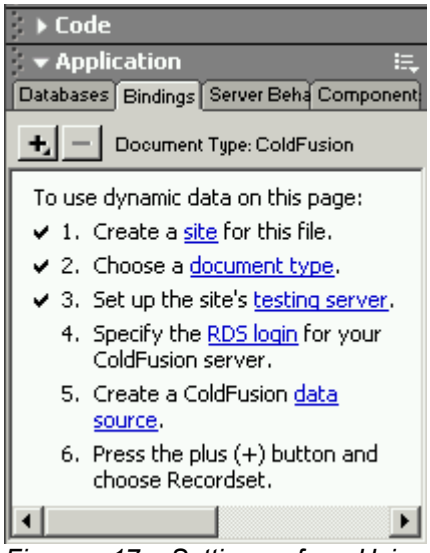


Figure 17 Settings for Using Dynamic Data

As you can see, the next step is to specify the RDS login for your ColdFusion server. After clicking on the ***RSD login*** link, enter the password to access the ColdFusion server:

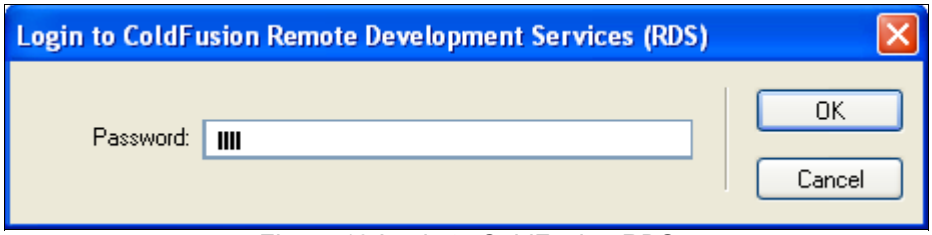


Figure 18 Login to ColdFusion RDS

If the ***mx_shop*** system DSN that you have created previously is found, it will be displayed in the **Databases** tab and the 4th and 5th steps of the **Bindings** tab list will be also checked. If not, click on the ***data source*** link and you will be redirected on a Internet browser window where you should:

1. Enter the ColdFusion server password.
2. Select the **Data Sources** sub-menu from the **DATA & SERVICES** left menu.
3. In the ***Add New Data Source*** section enter `mx_shop` for the ***Data Source Name***, select Microsoft Access from the ***Driver*** menu and click on the **Add** button.

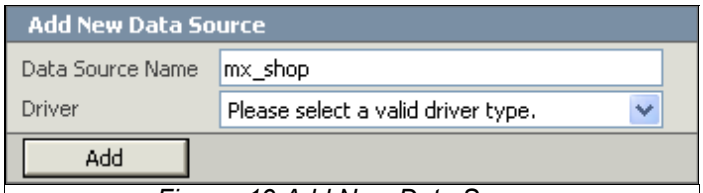
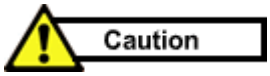


Figure 19 Add New Data Source

Go back in Dreamweaver MX and click on the **Refresh** button:

Figure 20 The Refresh Button

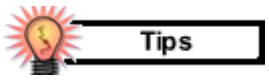
Normally, the **`mx_shop`** database should appear in the **Databases** list.



After creating the connection to the database, **Macromedia Dreamweaver MX** generates a folder, named **_mmServerScripts**. This folder contains some scripts that read information about your database and supply it to the Dreamweaver interface. When publishing your site, you have to delete this folder from the production server because it allows an attacker to gain unauthorized access to the database.

Using MX Kart

Now that you have the database, site, and connection set up you will be introduced to MX Kart's most common features. In this section you will create listing of products, and a shopping cart feature. The site will be implemented using MX Kart transactions and triggers.



Generally, a transaction is an interaction with a database. Triggers are events associated with a particular transaction and occur before or after the transaction, or when an error takes place, depending on your logic structure. Transactions and triggers actions which are integrated into the **Dreamweaver Server Behaviors**.

Creating a Product List

Open the *index.php/index.cfm* file. Before building the shopping cart, we recommend that you apply the **Update Kart Version** command to your site (**Application** panel > **Server Behaviors** tab > **Plus** > **MX Kommerce** > **Advanced**). If you are using MX Kart for the first time, this command generates all the files needed for the MX Kart behaviors to work correctly. If you already used an earlier version of MX Kart to build web applications, this command will update the necessary files in those applications.

To create the product list, you will first need to create a recordset to grab all of the information you are going to display. To create a new recordset:

1. In the **Bindings** tab of the **Application** panel, press the "+" button and select the **Recordset (Query)** option.
2. In the name field insert *rsProducts*
3. Select the **kart_conn** database connection.
4. In order to display data from the two tables you must create an advanced recordset. Click on the **Advanced..** button and write the following SQL query.

```
SELECT                                                    *
FROM                                                products_prd
LEFT    JOIN    categories_ctg    ON    products_prd.idctg_prd    =
categories_ctg.id_ctg
ORDER BY id_ctg ASC
```

5. Click **OK**.

The index.php/index.cfm page will contain a list of all the products, their categories, and a link for adding the product to the shopping cart. For displaying the recordset results, add a table (**Insert** bar -> **Tables** tab) with two rows and three columns.

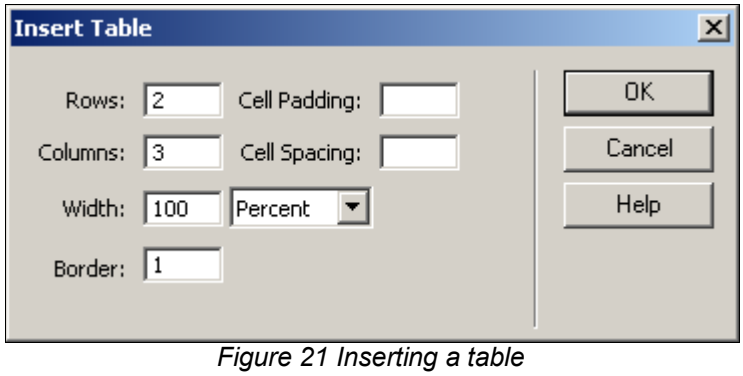


Figure 21 Inserting a table

- In the first table row write: “Category”, “Name” and “Price” in the three cells.

For a more friendly display you could configure the first row as header (by checking the ***Header*** box from the cells inspector).

- Go to **Bindings** and drag-and-drop the ***name_ctg*** (into the cell under the Category label), ***name_prd*** (into the cell under the Name label) and ***price_prd*** (into the cell under the Price label) dynamic fields from the *rsProducts* recordset (they have little lightening bolt icons next to them).

- Select the entire second row (select the first cell of the row then shift click on the last cell of the row) and apply a ***Repeat Region*** server behavior from the **Application** panel->**Server Behaviors** tab. When the dialog box opens, Select ***All records*** in the ***Repeat Region.*** Click **OK**. This way, all the products will be displayed and not just the first one.

  If you want to display the products in a more organized and visible manner (grouping them by categories) you can select the ***name_ctg*** field apply a **Show If Field Has Changed** server behavior. Access this behavior from the **Application** panel-> **Server Behaviors** tab->+->**MX Kollection-> Conditional Regions**. This way the category name will be displayed only once for the products belonging to the same category. Be sure to select the correct Field before you click **OK**.

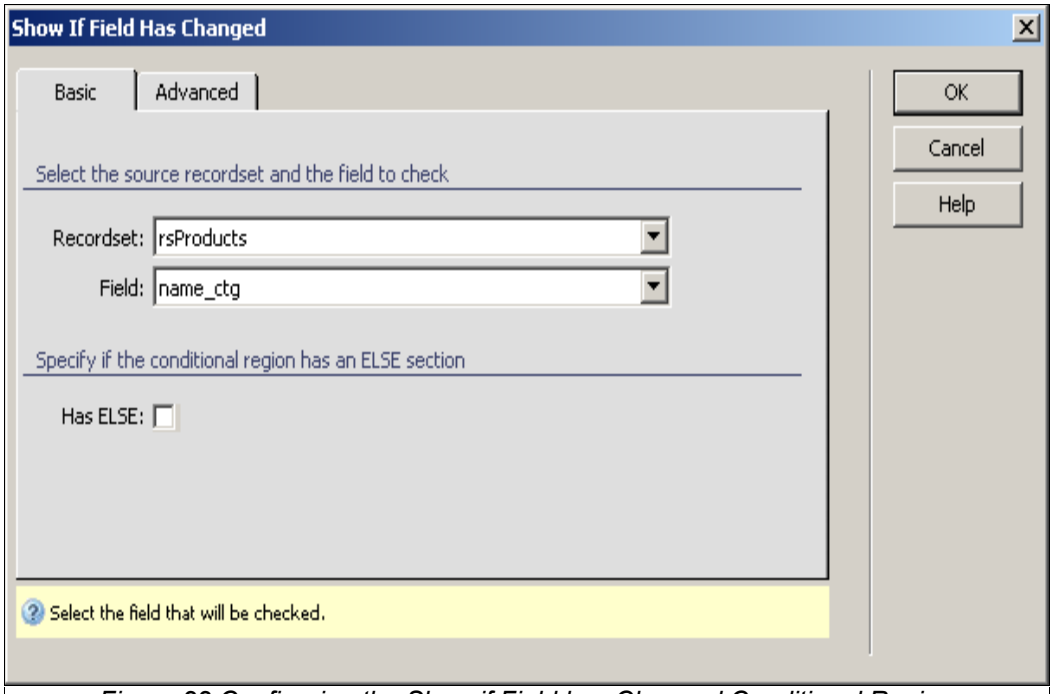


Figure 22 Configuring the Show if Field has Changed Conditional Region

A common item used to attract online shoppers is the display of a discounted price with the new lower price next to the original higher price with a strike-through line. To do this, you will have to duplicate the product price.

- Insert the ***price_prd*** dynamic field one more time next to the original price in the third column (it's under the **Bindings** tab).
- Select the first price field, right click and choose ***Style*** > ***Strikethrough***.

This is how the product list table should look like at this point:

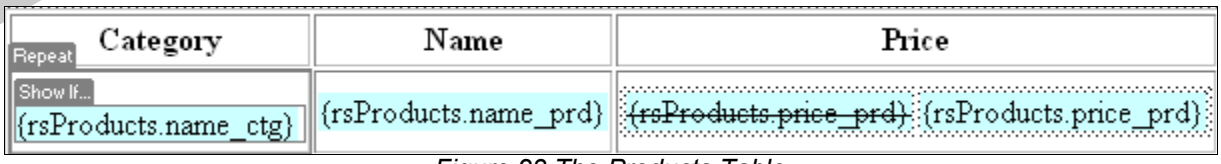


Figure 23 The Products Table

Applying the Show If Discounted Product Server Behavior

The old strike-through price should only be displayed if a discount exists. For that you must select the old price field (the one WITH the strike-through) and apply the **Show If Discounted Product** Server Behavior from the **Application** panel > **Server Behaviors** tab > **Plus(+) >MX Kommerce > Show if discounted Product**. The ***Product Price*** user interface field will be automatically filled in with the ***Product Primary Key Value***. Select the product ID field from the *rsProducts* recordset and click **OK**:

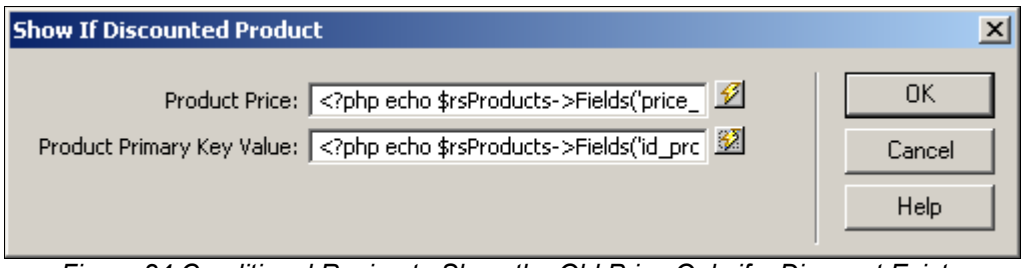


Figure 24 Conditional Region to Show the Old Price Only if a Discount Exists

Applying the Show Product Price Server Behavior

The current list presents the product prices in an unformatted way.

In order to format the old price field to display the currency and other formatting rules, select it and apply the **Show Product Price** Server Behavior (**Application** panel > **Server Behaviors** tab > **Plus (+)** > **MX Kommerce** > **Show Product Price**). Because you selected the price field it will be automatically inserted into the *Price* field of the **Show Product Price** server behavior interface.

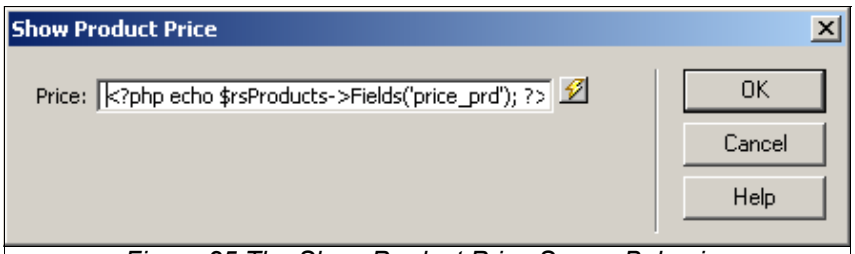


Figure 25 The Show Product Price Server Behavior

Save the `index.php` file, upload it to the server and test it with the browser by using the F12 key.

You should see the regular dollar formatting rules applied on all your visible product price fields. USD is the default currency in the database provided.

Applying the Add to Kart by Link Server Behavior

In this section you will add a link to the product table which will add the product to the site user's shopping cart.

1. In the product list table, add a new column to accommodate the new Add to cart link. Right click in the price cell and select ***Table->Insert Rows or Columns***

2. Write “Add to cart” in the new column on the second row, and select ***No wrap*** in the property inspector.
3. Select the “Add to cart” text and apply the **Add to Kart By Link** Server Behavior from the **Application** panel > **Server Behaviors** tab > **Plus (+)** > **MX Kommerce > Add to Kart By Link**.

The next step will be to configure the dynamic bindings (by pressing the “lightning bolt” button and selecting from the recordset the corresponding fields – See image below) for the attributes that will be passed to the Kart recordset.

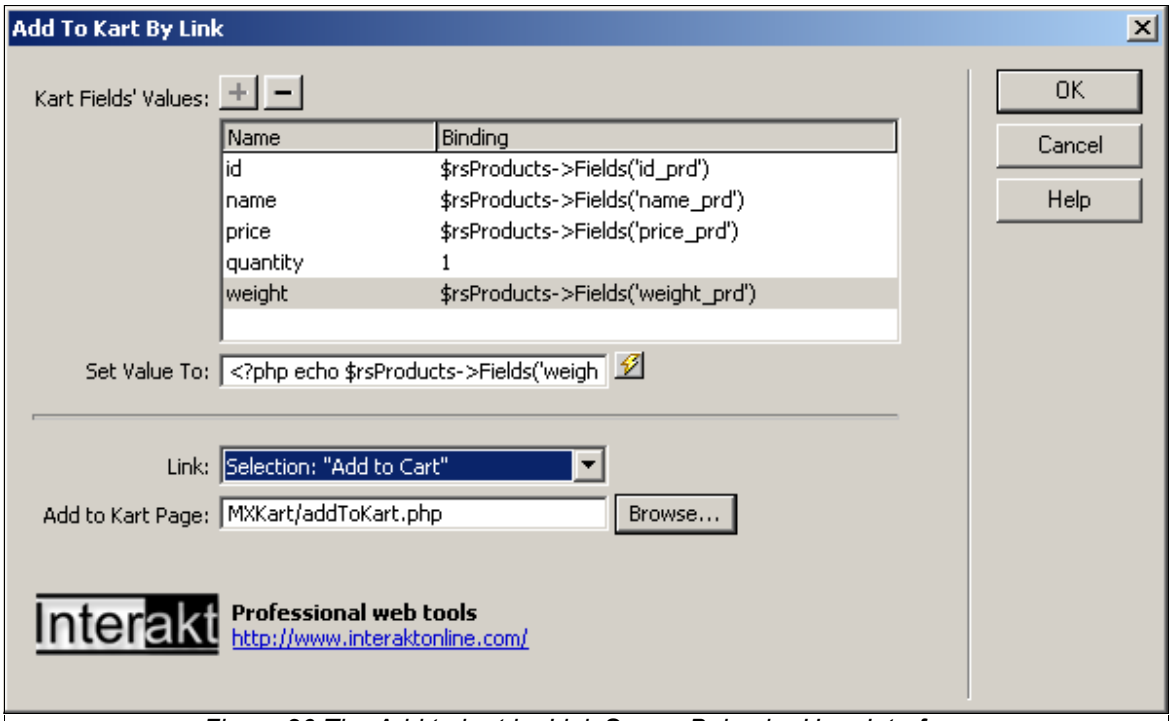


Figure 26 The Add to kart by Link Server Behavior User Interface

The value to be entered for the "***quantity***" parameter should be numerical. This is the quantity of the selected product to be added to the shopping cart when the ***Add to cart*** link is clicked.

4. In the ***Add to Kart Page*** text field you should enter the name of the page that actually executes the add to cart transaction. In this case you should not change the page that is already entered – `MXKart/addToKart.php`, since this page behaves correctly and is included with the **MX Kart** kit.
5. Click OK.

Creating the Product Detail Page

Configuring the Recordset

1. Open the `prodDetail.php` page which will allow the customer to select the desired product properties and quantities and then to add the product to cart.
2. Create a new recordset named *rsDetails*. This recordset will contain all the fields from the *products_prd* table for a certain product. If you want to display the corresponding category name (from the *categories_ctg* table) as well then you should create an advanced recordset.

To create a new recordset go to the **Bindings** tab in the **Application** panel and press the "+" button and select the **Recordset (Query)** option.

3. If the interface is different than the one shown below you are probably using the “simple” view. Click on the **Advanced** button. Now configure the recordset as in the picture below. You may wish to copy and paste from the text under the image below.

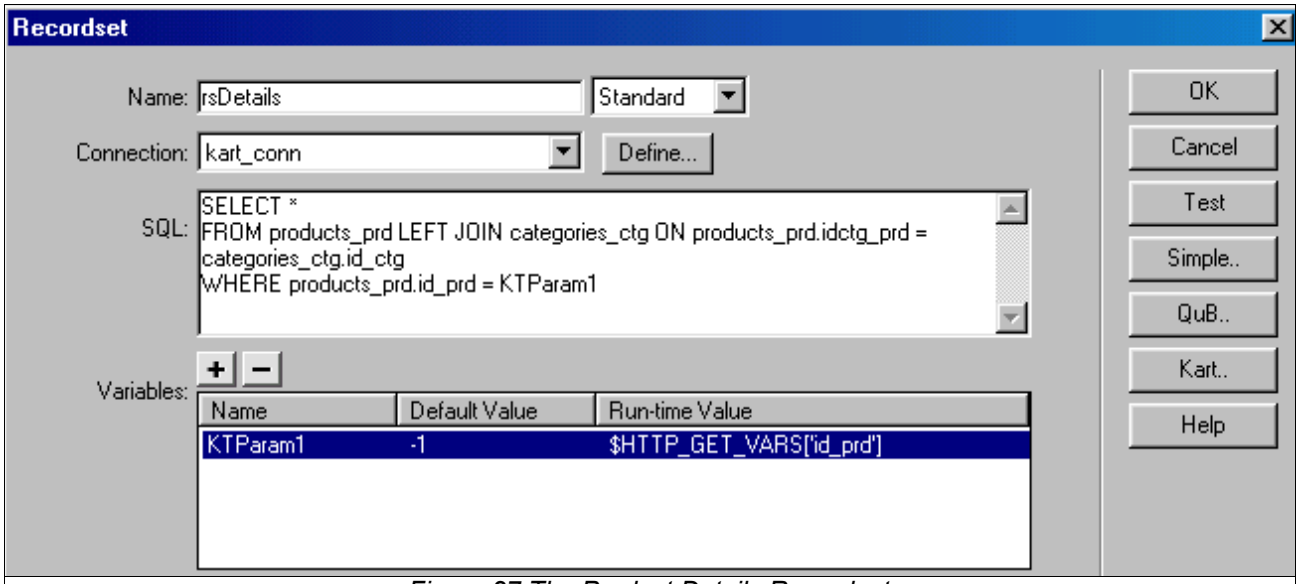


Figure 27 The Product Details Recordset

The SQL query should be the following:

```
SELECT *
FROM products_prd LEFT JOIN categories_ctg ON products_prd.idctg_prd = categories_ctg.id_ctg
WHERE products_prd.id_prd = KTParam1
```

As you can see we created a variable named ***KTParam1*** (you can choose another name) that will pass the ***id_prd*** value obtained by a GET procedure to the SQL.

4. To create this variable press the + in the Variables section**.** Use these settings:

Name: KTParam1

Default Value: -1

Runtime Value: $HTTP_GET_VARS['id_prd']

The recordset will be filtered after the ***id_prd*** field. See image below:

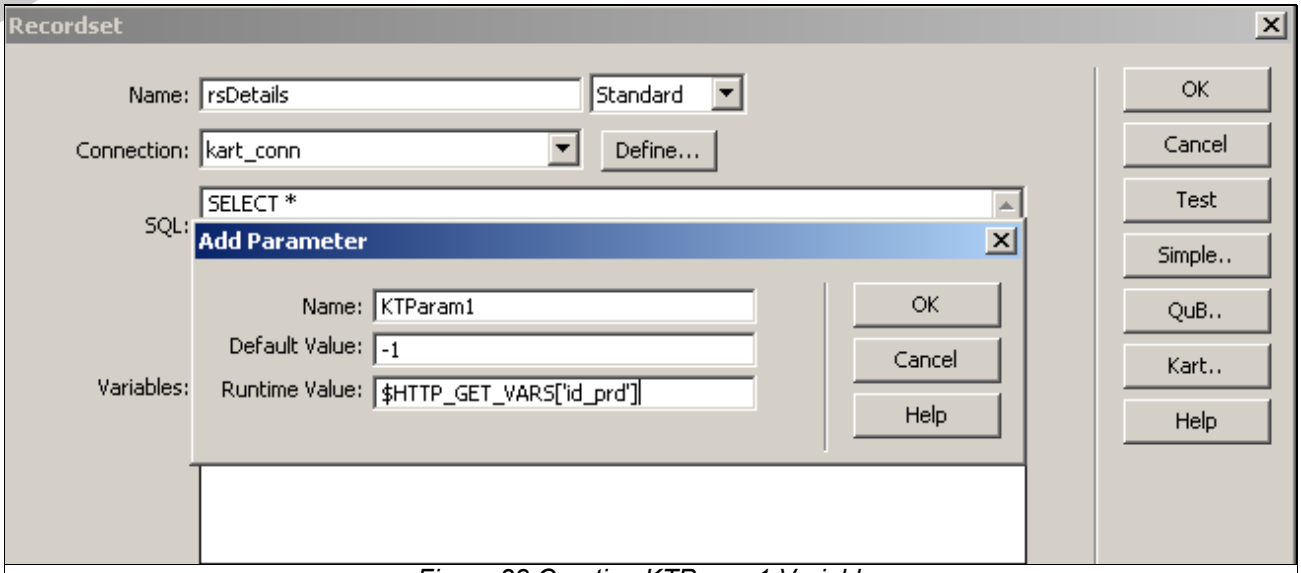


Figure 28 Creating KTParam1 Variable

5. Click **OK** when done, then click **OK** again.

The next step is to insert a table where all the product details will be displayed.

- Insert a table having 6 rows and 2 columns (**Insert** bar > **Common** tab).
- In the first column on each row write: "Category", "Product name", "Description", "Price" and "Options". Align the first column to the right, by selecting the entire column and setting the ***Horiz*** field to ***Right,*** then check the ***Header*** box.
- For the first four rows drag-and-drop the corresponding dynamic values from the *rsDetails* recordset in the **Application** panel-> **Bindings** tab.
- For the ***Price*** row insert two price dynamic values as you did on the `index.php` page. The old price will be displayed with a strike-through only if the current product is discounted This should be formatted to display currency. To do this you should apply the **Show Product Price** AND **Show If Discounted Product** server behaviors in the same manner as explained above (Figures 14 and 16).

  The table should look as in the following image:

Category	{rsDetails.name_ctg}
Product Name	{rsDetails.name_prd}
Description	{rsDetails.description_prd}
Price	Show if Discounted Product ~~{rsDetails.price_prd}~~ {rsDetails.price_prd}
Options	

Figure 29 The Product Detail Table

- Save the `prodDetail.php` file.

Applying the Add to Kart by Form Command

Before implementing the **Add To Kart By Form** Command, you should understand how the **Add To Kart By Link** and **By Form** behaviors work:

- If the customer uses an "Add to Cart" link that was implemented with the **Add To Kart By Link** Server Behavior the **Redirect If Properties Not Set** trigger (included on the

addToKart.php page from the **MX Kart** folder) will check if the selected product has properties that need to be specified before adding it to the cart. For example, when purchasing a keyboard, the site user will need to select the type of connection (USB, wireless, etc.) before the item is added to their cart.

- If the product has extra properties, the customer will be redirected to the product detail page where he will be able to choose the product properties and then use the **Add To Kart By Form** Button.
- If the product does not have properties the "Add to Cart" transaction will continue, and the site user will remain on the product list page.
- If the customer uses an "Add to Cart" button that was implemented with the **Add To Kart By Form** Command from the *prodDetail.php* page the trigger will add the selected properties and will modify the product price according to the selection. Then the "add to cart" transaction will continue.

1. In the *prodDetail.php* page, click the cell corresponding to "Options" and apply the **Add To Kart By Form** Command from the **Insert** Panel->**MX Kart** tab.

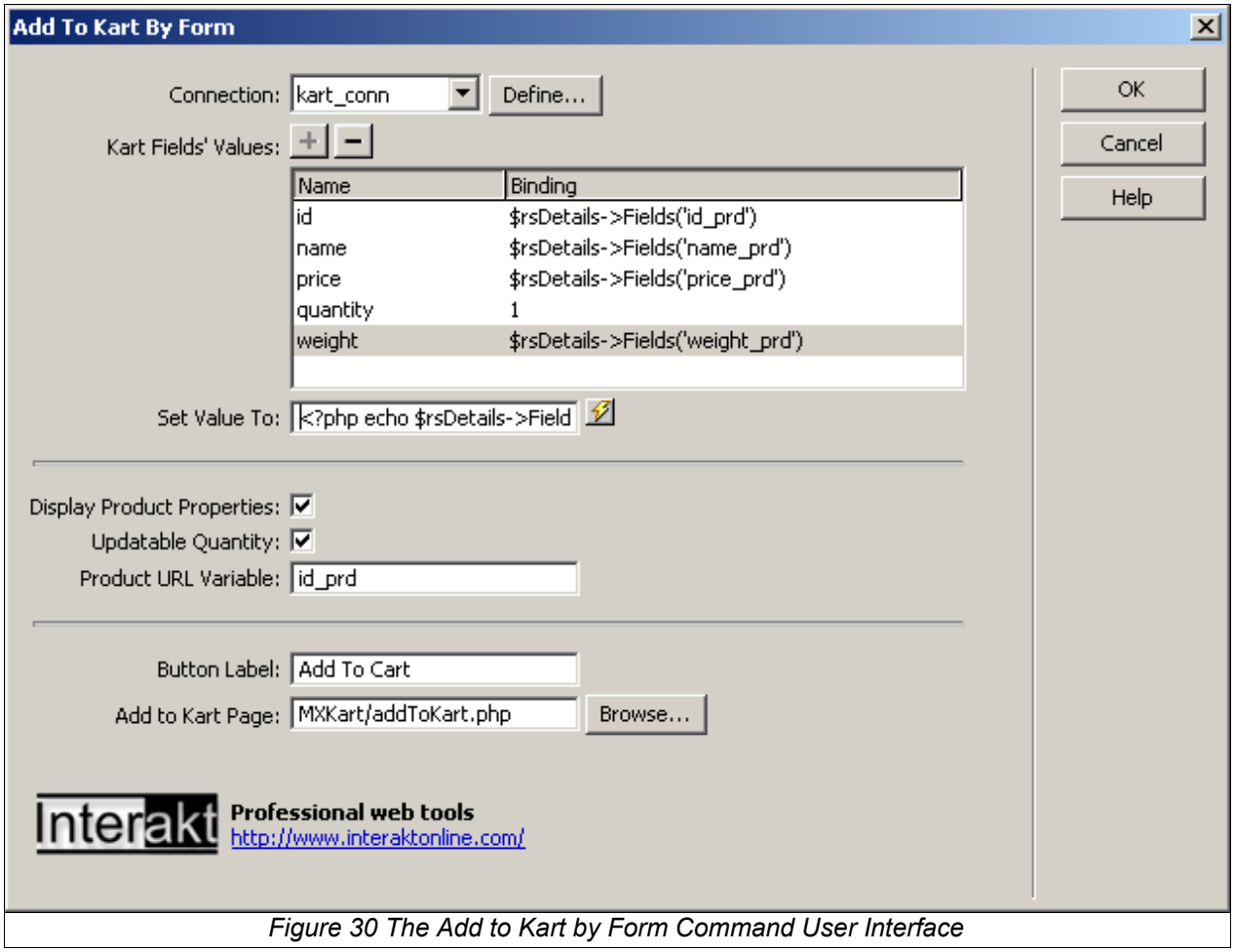


Figure 30 The Add to Kart by Form Command User Interface

2. Fill in the configuration window as follows:

- ***Connection*** – select the name of the database connection to which the transaction generated by this command is registered.
- Leave the list with the default parameters that are to be stored in the session recordset unchanged.

- ***Set Value To*** – this field will allow setting a dynamic value for each parameter. You will simply select the parameter and choose a dynamic value for it by clicking on the "lightning bolt" icon. For the "***quantity***" parameter the value to be entered should be numeric. This will be the default quantity that will appear in the "add to cart" form.
- ***Display Properties*** – when checked this box allows the product options (properties) to be displayed
- ***Updatable Quantity*** – when checked this box enables the ***quantity*** field to be editable
- ***Product URL Variable*** – in this text field you should enter name of the variable that was sent by URL (the same one that was used as filter when creating the recordset) – in this case, enter ***id_prd***
- ***Button Label*** – specify the label of the button that adds items to the shopping cart.
- ***Add to Kart Page*** – in this text field you should enter the name of the page that actually executes the add to cart transaction (in this case, you should not change the link that is already entered – `MXKart/addToKart.php`). Use the **Browse** button to select the page.

3. Click **OK**.

4. In the last row of the table write ***Product list*** and make a link to the `index.php` page.

   The Dreamweaver `prodDetail.php` page should look similar to the one in the image below:

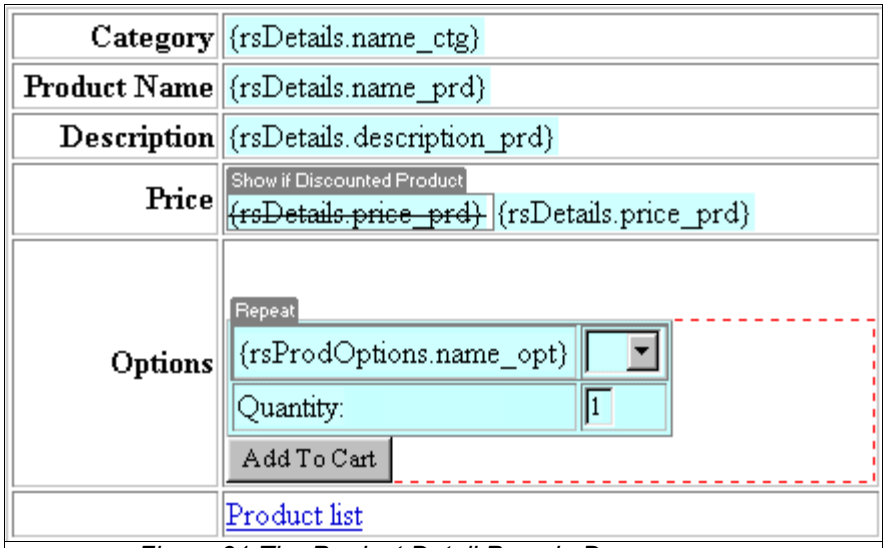


Figure 31 The Product Detail Page in Dreamweaver

5. Save the `prodDetail.php` file and upload it to the server.

Configuring the addToKart.php page

This section explains how the `addToKart.php` page included in the **MX Kart** folder works. When loaded, this file will execute the "Add to Kart" transaction. This transaction will add the selected product to the cart or will redirect the customer to the product detail page if the product has options and have not been chosen yet.

The triggers registered to this transaction are the following:

- **Starter** - the transaction will start if the customer clicks an "Add to Kart" link generated by **MX**

Kart

- **Redirect If Properties Not Set** If the customer used an "Add to Kart By Form" button, the trigger will add the selected properties and will modify the product price according to the selection. Then the transaction will continue
- If the customer used "Add to Kart By Link", the trigger will check if the selected product has properties
  - If the product has properties the customer will be redirected to the product detail page where he will be able to choose the product properties and then use the Add To Kart By Form Button
  - If the product does not have properties the transaction will continue.
- **Redirect -** after the product is added to the Kart recordset, the customer is redirected to the page he came from (to the referrer).

1. Open the `addToKart.php` page (inside the `MXKart` folder) and double click on **Redirect If Properties Not Set** from the server behavior list.

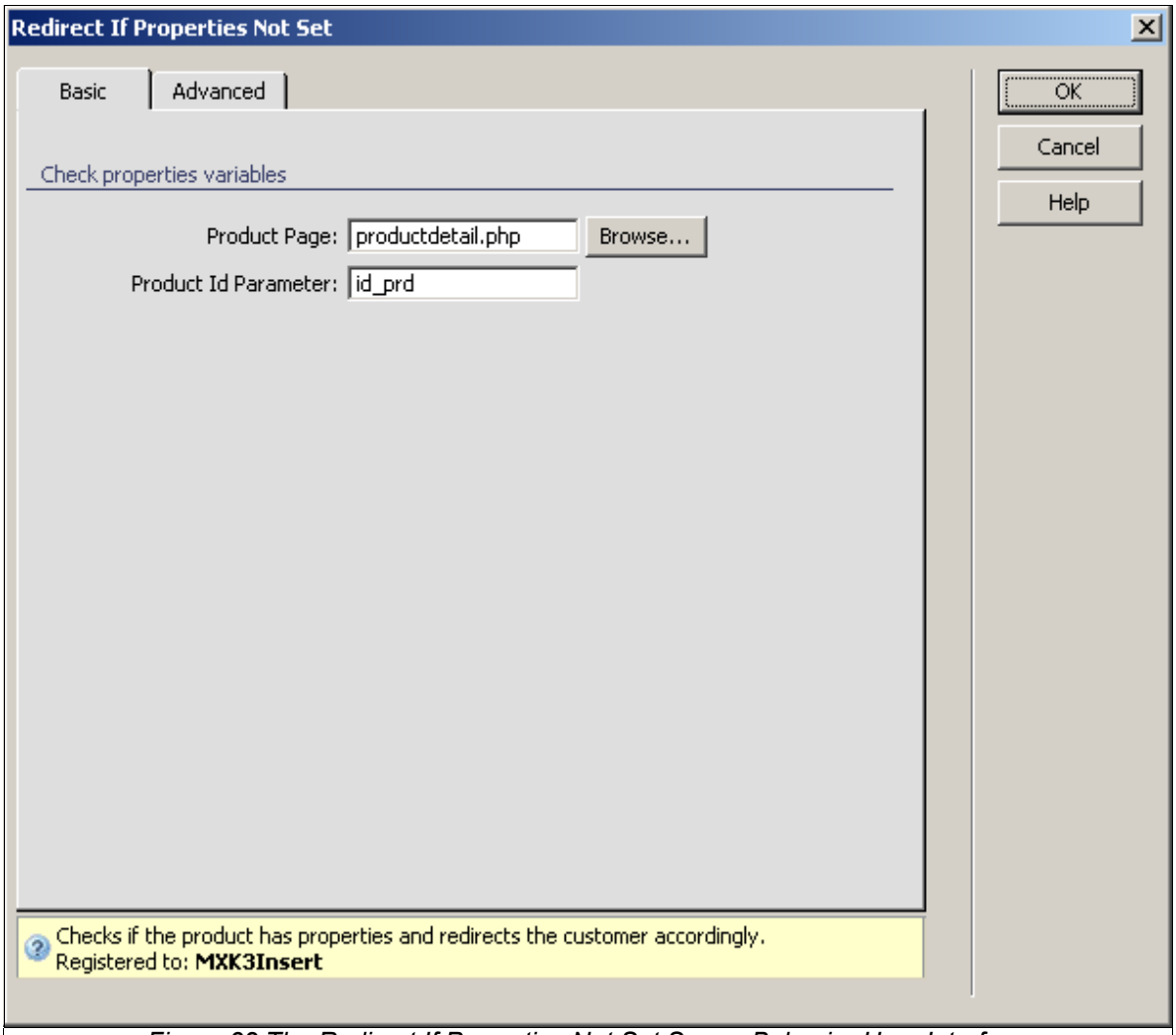


Figure 32 The Redirect If Properties Not Set Server Behavior User Interface

2. In the interface only change the field related to the ***Product Page***. In the Product Page text field, browse to the page where you applied the **Add To Kart By Form** command. In this case you will browse to the `prodDetail.php` page.

3. Next double-click on the **tNG(InsertIOntoKart)** transaction. The following interface will be displayed:

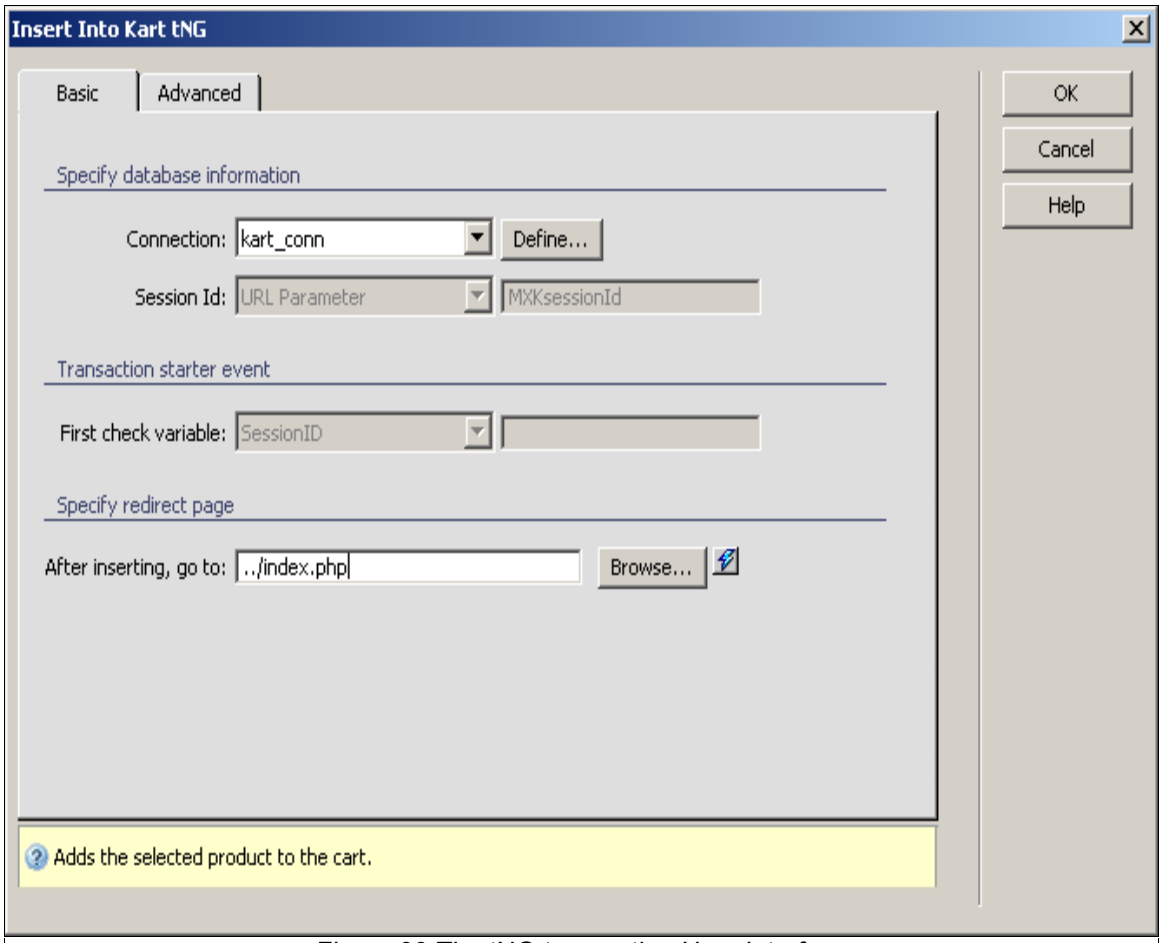


Figure 33 The tNG transaction User Interface

You can change the page your customer is redirected to by browsing to a different page in the A***fter inserting, go to*** field. This will be the page loaded after an item is added to the shopping cart. The default redirect is to the page the "add to kart" transaction was applied but it could instead be to the products list page.

Creating the View Cart Page

The `viewKart.php` page will display the entire contents of the cart, not just the total price and ***Checkout*** link.

Applying the Create Shopping Kart View command

1. Open the `viewKart.php` page and apply the **Create Shopping Kart View** command from the **Insert bar** > **MX Kart** tab.

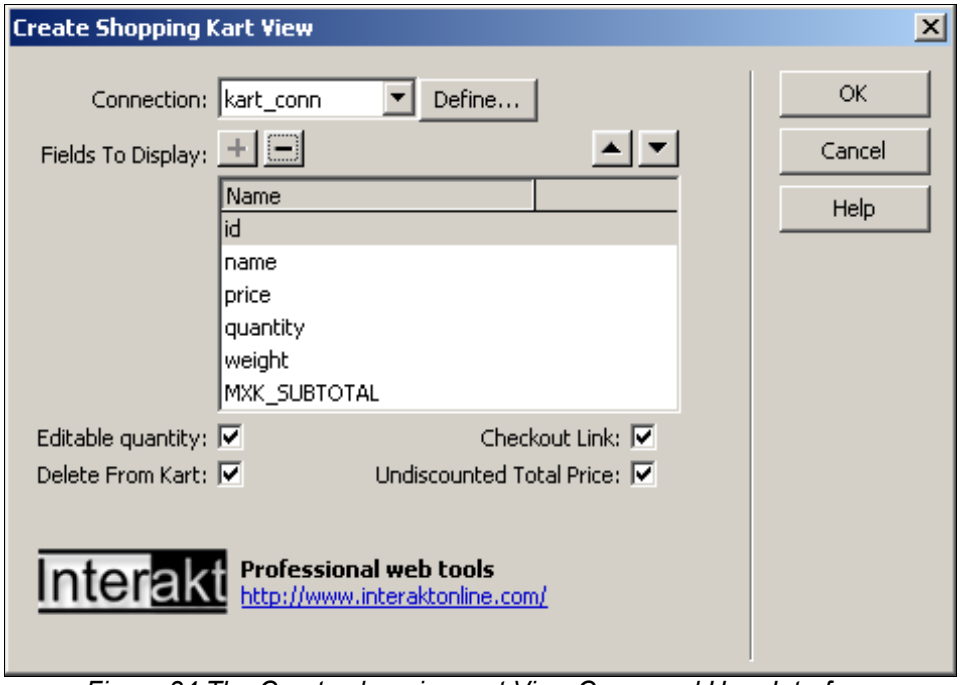


Figure 34 The Create shopping cart View Command User Interface

2. Specify the fields in the configuration window:
   - ***Connection*** – in this drop down menu select the name of the database connection the transaction is registered to. The **Define** button will open a configuration window allowing you to create another connection.
   - a list with the default parameters that are to be stored in the session recordset. You can add or delete parameters by using the "+" or "-" parameters and you can also set the parameters' display order in the cart by using the up/down arrows.
   - ***Editable quantity*** – when checked it will make the **Update** button visible, the quantity field editable and it will introduce an update transaction for it.
   - ***Checkout Link*** – when checked it will insert a ***Checkout*** link that will start the checkout process.
   - ***Delete From Kart***– when checked it will display a ***Delete*** link that removes the corresponding items from the cart .
   - ***Undiscounted Total Price***– when checked it will display the regular ("undiscounted") price next to the discounted one.

   The code generated by this command is a repeated region that adds dynamic values for the fields in the shopping cart recordset. Besides these values the **Create shopping cart View** Command also includes a ***Total Price*** dynamic value.

The Dreamweaver `viewKart.php` page should look similar to the one presented in the image below:

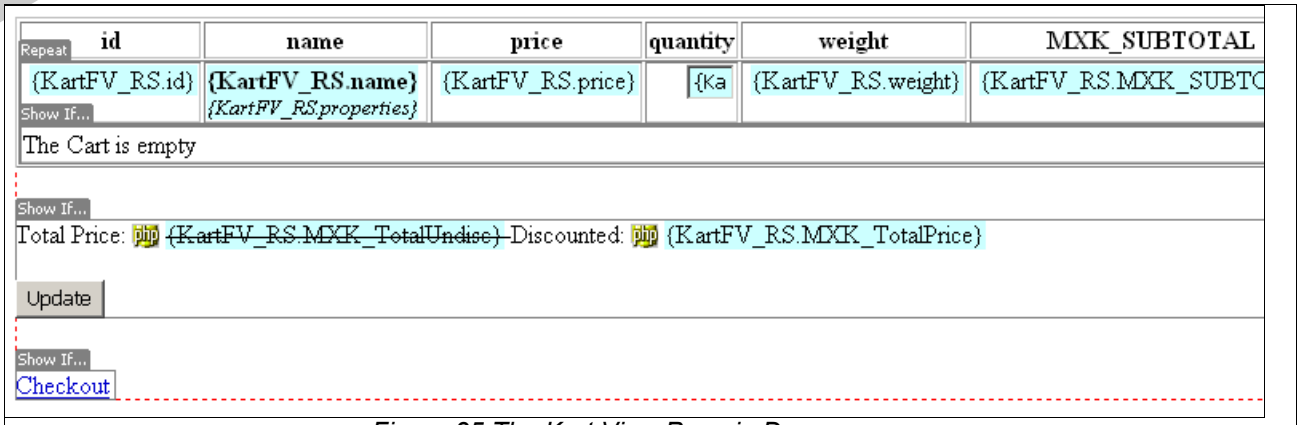


Figure 35 The Kart View Page in Dreamweaver

The checkout procedures are outside the scope of this tutorial. To create the Checkout section, see the **MX Kart Tutorial** also included in your installation package.

Caution

The **Create Shopping Kart View** command will include a **Redirect To Page** server behavior registered to the Delete transaction. The redirection is made to the same page (the referrer) and you should keep in mind not to check the ***Keep URL Parameters*** check box in the **Redirect To Page** server behavior user interface because you risk entering an infinite loop. The reason is that the Delete transaction will try to execute each time the `MXKDel` URL parameter is passed to the page on redirect.

3. Save the `viewKart.php` file and upload it to the server. Test it with the browser by using the F12 key.

Congratulations on finishing the **Getting Started with MX Kart** tutorial. You have now learned how to build a product list, shopping cart, and dynamic behaviors to modify the kart contents. These features form the core of an e-commerce application. In the next guide, **MX Kart Tutorial,** you will build upon the features here, and create a full e-commerce site.

Where to Go From Here

Now that you have been introduced to several popular features, you can move on to the more specific tutorials for getting the most out of MX Kart. The more detailed **MX Kart** tutorial will provide you with instructions for completed a full e-commerce site.

Topics in the **MX Kart Tutorial** include:

- User Login/Logout
- Applying Discounts and Coupons
- Applying taxes per tax-zone
- Multiple currency support
- Configuring a Payment Gateway (PayPal, WorldPay, etc.)

Through these two tutorials, you will be able to create complex e-commerce sites in very little time, and without writing any code. Enjoy using MX Kart, and let us know how you find our products and tutorials by e-mailing us at products@interaktonline.com.