

Adobe

Adobe Developer Connection

[Sign in and join ADC](#) [Newsletters](#) [Feedback](#) [RSS](#)

Dreamweaver Article

Developing a web application with Dreamweaver

Charles Nadeau

Adobe

In this tutorial, you'll learn how to use Adobe Dreamweaver CS3 to quickly begin developing database-driven web applications. You'll learn how to retrieve data from a database and display it on web pages. You'll also use Dreamweaver to create a form that enables visitors to leave comments and then saves their comments in a database.

Requirements

You must set up your Dreamweaver work environment for application development before you can complete this tutorial. For instructions, complete one of the setup tutorials from the following list:

Dreamweaver CS3

[Try](#) [Buy](#)

- [Setting up a PHP development environment for Dreamweaver](#)
- [Setting up a ColdFusion development environment for Dreamweaver](#) (coming soon)
- [Setting up an ASP development environment for Dreamweaver](#) (coming soon)

These setup tutorials teach you how to complete the following tasks:

- Install and run a web server as well as a PHP, ColdFusion, or ASP application server on your local computer
- In the case of PHP, install the MySQL database server on your local computer
- Define a Dreamweaver site for the sample website used in this tutorial

- Connect to the sample database used in this tutorial
- The screen shots in this tutorial show PHP dialog boxes in Windows. However, you can complete the present tutorial in any one of the three development environments—PHP, ColdFusion, or ASP—in either Windows or on the Macintosh.

Review your task

Your task is to build the following dynamic pages for the fictional Cafe Townsend website:

- A page that displays the comments saved in a database.
- A page that enables site visitors to send comments to the company, where they are saved in the database.

Figure 1 shows the page that enables Cafe Townsend staff to view the comments in the database.

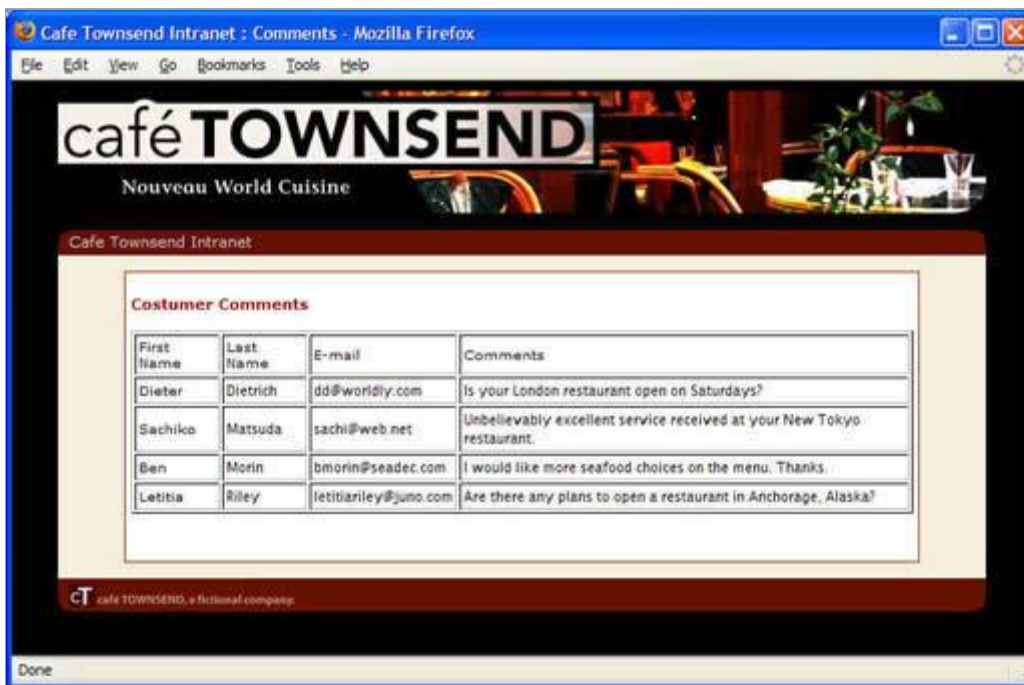


Figure 1. This page enables you to view customer comments.

Figure 2 shows the page that enables visitors to send their comments.

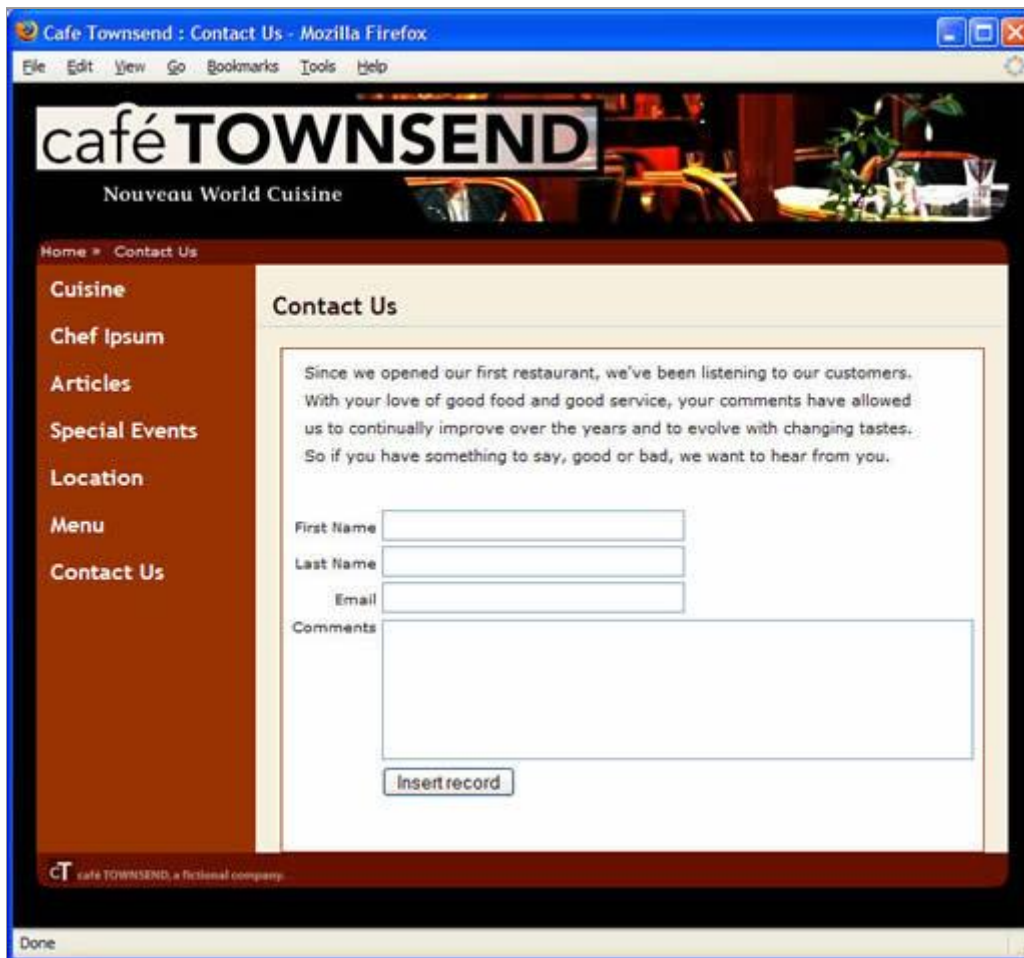


Figure 2. This page is for customer feedback.

Open a document to work in

A good starting point for developing a database application is to create a page that lists records stored in the database. In the application you're building in this tutorial, you'll create a dynamic web page that lists information drawn from the customer comments table of the database. Later, you'll create another dynamic web page that enables a customer to insert comments or questions directly into the database.

Start by locating the documents you'll work with as you create these pages:

1. Select Window > Files to open the Files panel (see Figure 3).
2. In the Site pop-up menu, select the site you defined during the setup tutorial.

If you followed the naming suggestion in the setup tutorial, the site is called *Web app tutorial*. For more information, see the setup tutorials listed in Requirements.

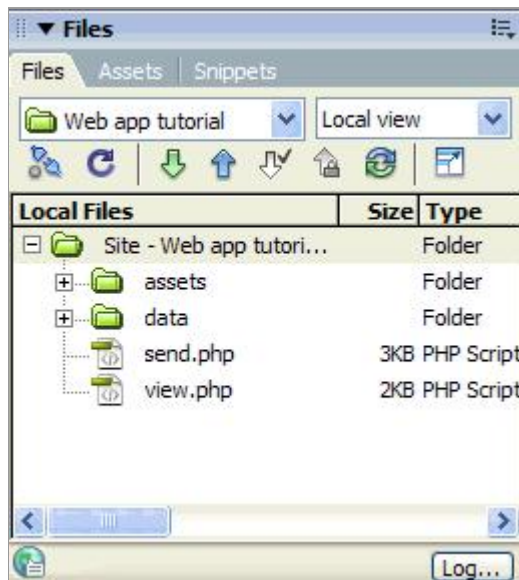


Figure 3. The Files panel of the Site pop-up menu.

3. In the Files panel, double-click the file named view.php, view.cfm, or view.asp (depending on your setup).

The document opens in the Document window.

4. If you are currently viewing the document in Code view, click Show Design View (labeled Design) or Show Code and Design Views (labeled Split) in the Document toolbar.

Define a recordset

Now you'll create a recordset to select the data you want to display. A recordset is a set of information extracted from a database by a database query. A database query is a way to request data from a database by using specified search criteria, usually in a language called SQL. You then use the extracted information as a source of content for your dynamic pages.

You don't need to know the SQL language to create a recordset in Dreamweaver, which provides an easy-to-use interface for creating simple SQL queries. If you do know SQL, Dreamweaver also lets you write your own queries if you want to do so.

You'll create a recordset that selects all of the values from the COMMENTS table in the tutorial database.

Note: Before you can complete this procedure, make sure you connect to the sample database as explained in the setup tutorial. For more information, see Requirements.

1. If you haven't already done so, double-click the file named view.php (or view.cfm, or

view.asp, depending on your setup) in the Files panel to open it in Dreamweaver.

2. Open the Recordset dialog box by selecting Window > Bindings to open the Bindings panel, and then clicking the Plus (+) button on the panel toolbar and selecting Recordset.

Note: If the dialog box you see looks more complex than the one in Figure 4, click Simple.

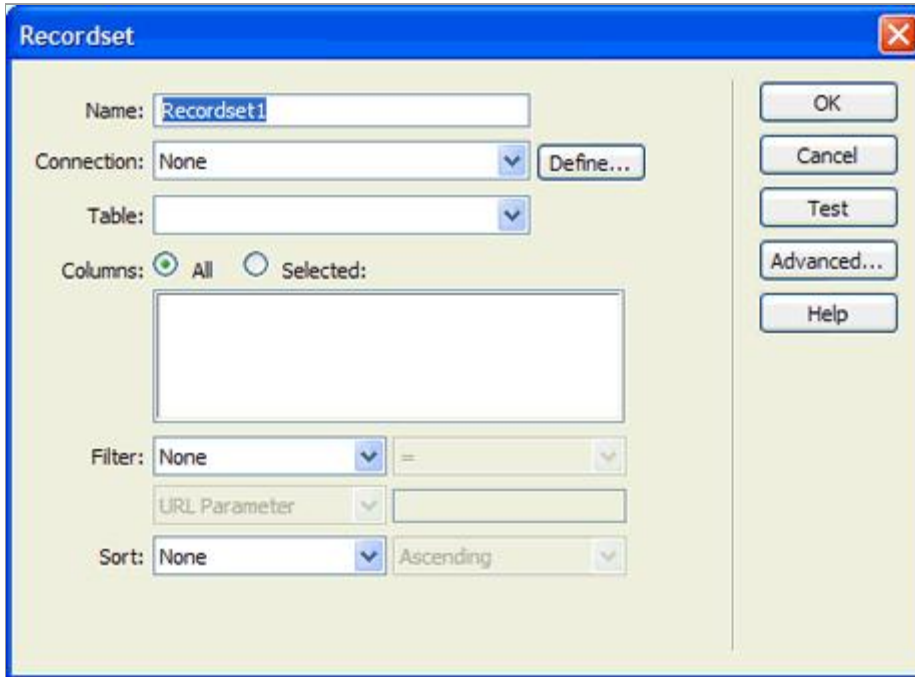


Figure 4. The Recordset dialog box.

3. In the Name text box, enter **rs_Comment**.
4. In the Data Source pop-up menu (ColdFusion) or Connection pop-up menu (PHP and ASP), select connTownsend.

The Recordset dialog box updates with data from the database (see Figure 5).

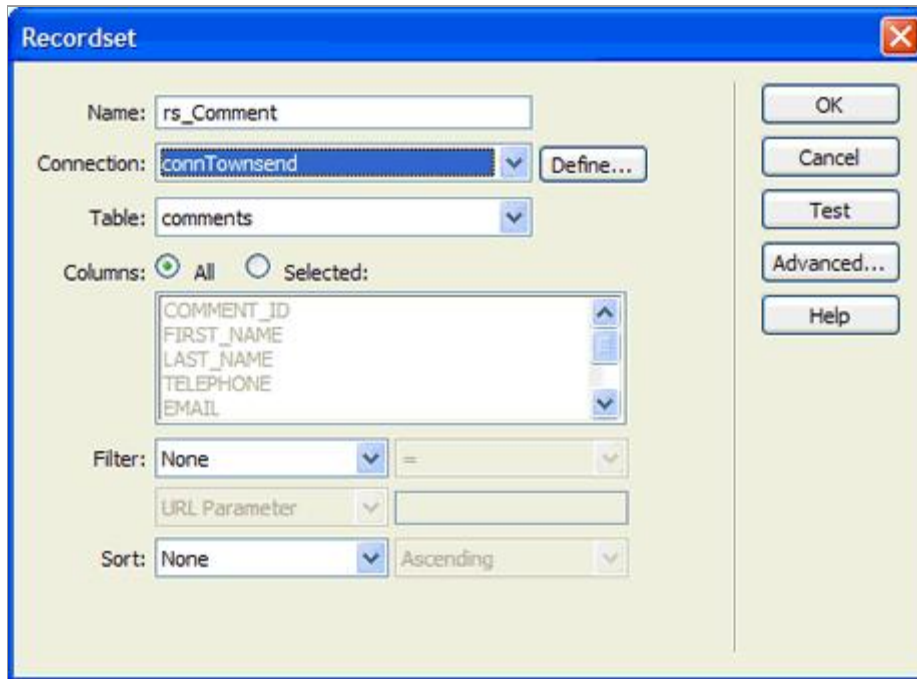


Figure 5. The updated Recordset dialog box.

5. In the Table pop-up menu, leave COMMENTS selected.
6. In Columns, make sure All is selected to select all columns of the table.
7. Leave the Filter pop-up menu set to None to select all rows of the table.
8. In the first Sort pop-up menu, select LAST_NAME, and in the second pop-up menu, select Ascending.

This step specifies that the retrieved records will be listed in alphabetical order by customers' last names.

9. Click Test to test the recordset.

The records from the database that match your recordset selection criteria are displayed in the Test SQL Statement window (see Figure 6). In this case, you selected all the data in the table.

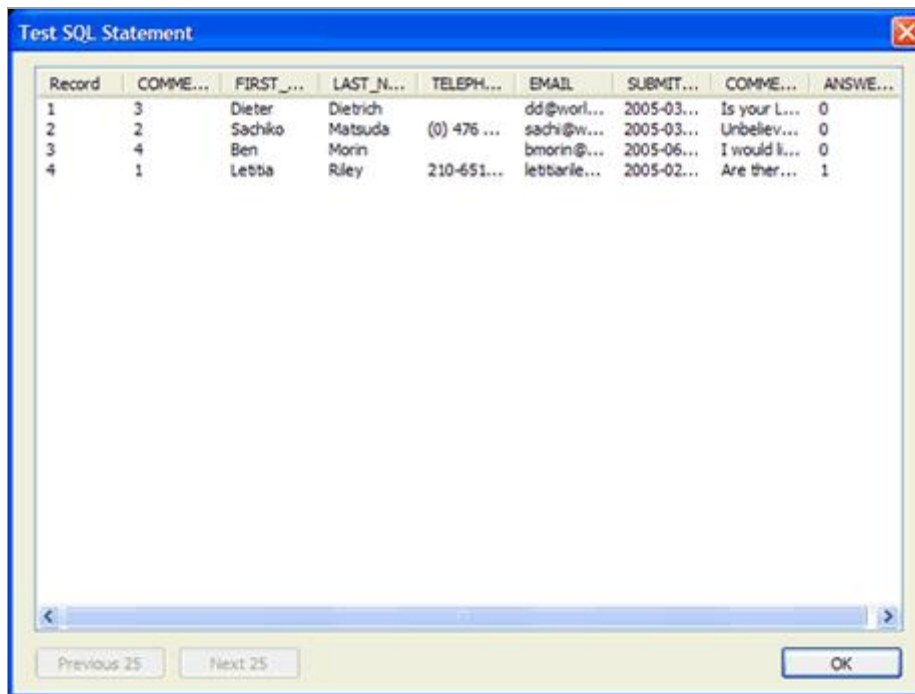


Figure 6. The Test SQL Statement window.

10. Click OK to close the Test SQL Statement window.
11. Click OK to close the Recordset dialog box and create the recordset.

The recordset appears in the Bindings panel (see Figure 7).

Tip: If you don't see all of the recordset fields (or columns) in the Bindings panel, click Plus (+) (Windows) or click the expander arrow (Macintosh) next to the Recordset (rs_Comment) item.

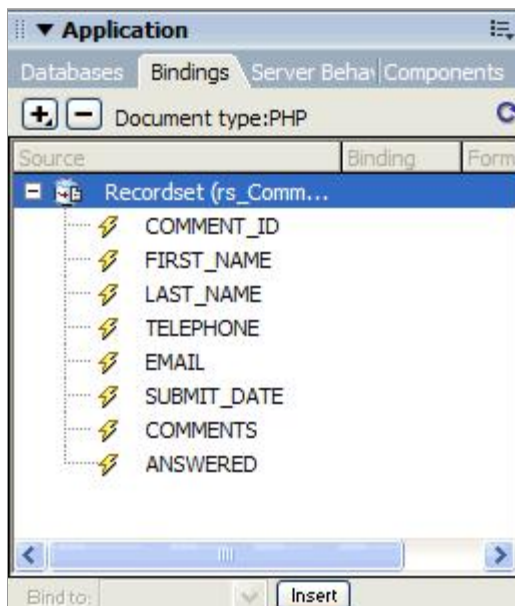


Figure 7. Recordset fields displayed in the Bindings panel.

Create a table to display the data

Next, you'll create a table on your page to display records in the database. The page will dynamically populate the table; you won't have to enter the information manually.

1. In the file named view.php (or view.cfm, or view.asp), place the insertion point in the blank line after the Customer Comments heading.
2. Select Insert > Table.
3. In the Table dialog box, set the following options:
 - In Rows, enter **2**.
 - In Columns, enter **4**.
 - Leave Table Width set to **100** percent.
 - In Border Thickness enter **1**.
 - In Cell Padding enter **2**.
 - In Cell Spacing, enter **2**.
 - In the Header section, select the Top option.

In the Summary text box, provide a textual description of the table for the benefit of visitors who use screen readers, as follows:

This table lists all of the comments that visitors to this site have entered into the database.

Figure 8 shows how the completed dialog box should look with the exception of the Header section. You should instead have the top option selected.

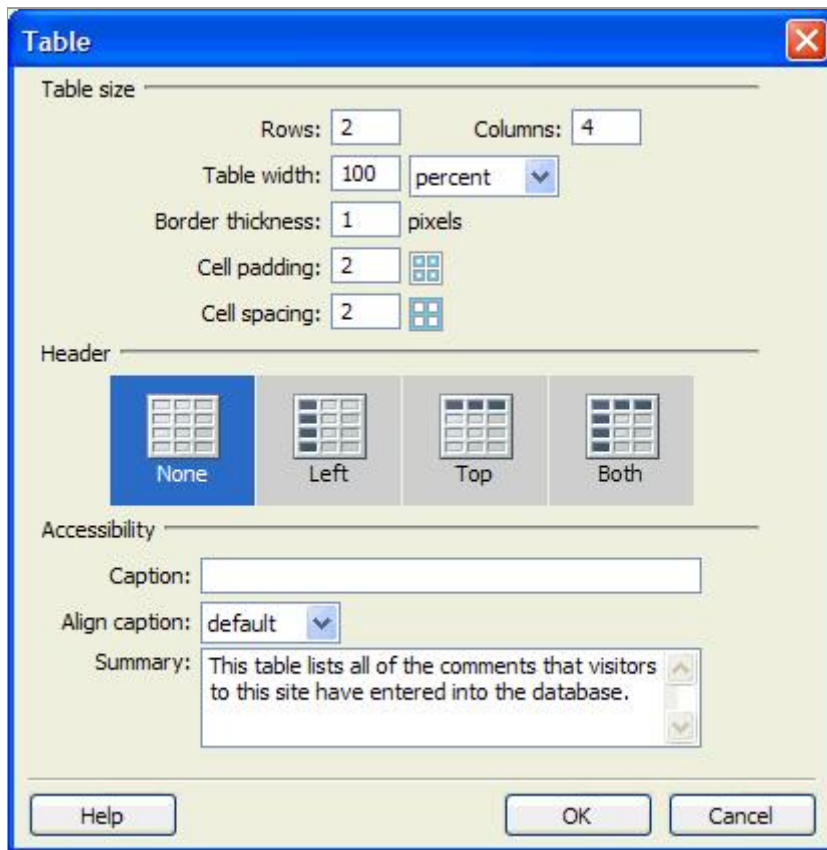


Figure 8. Setting table options and providing a summary in the Table dialog box.

4. Click OK.

Dreamweaver inserts the table in the document.

5. In the top row of the table, add labels for the table entries:

- In the first cell of the table, enter **First Name**.
- In the next cell, enter **Last Name**.
- In the next cell, enter **E-mail**.
- In the last cell, enter **Comments**.

Figure 9 shows how the table should look.

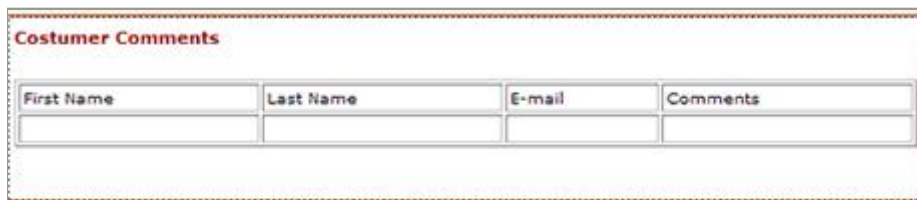


Figure 9. Labels inserted in the table.

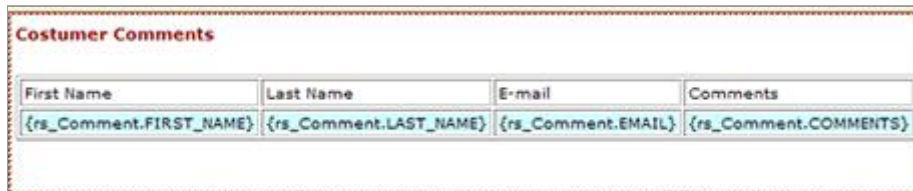
6. Save your page.

Add dynamic fields to the table

You're now ready to add the recordset fields to the table. This step is also known as *binding the data to the table*.

1. If it isn't already open, open the Bindings panel by selecting Window > Bindings.
2. Add the FIRST_NAME field to the table by placing the insertion point in the table cell below the First Name label and then, in the Bindings panel, selecting FIRST_NAME and clicking Insert.
3. Add the LAST_NAME field to the table by using a different technique: drag LAST_NAME from the Bindings panel to the table cell below the Last Name label.
4. Use your preferred technique (from either Step 2 or 3) to add EMAIL and COMMENTS (not COMMENT_ID) to the table.

Figure 10 shows how the table should look.



First Name	Last Name	E-mail	Comments
{rs_Comment.FIRST_NAME}	{rs_Comment.LAST_NAME}	{rs_Comment.EMAIL}	{rs_Comment.COMMENTS}

Figure 10. Recordset fields added to the table.

5. Save your page.

Create a repeated region

The table you created contains only one row for data. To display all the records in the recordset, you must define the table row as a repeated region. When viewed in a browser, the row will be repeated for each record in the recordset, so that all the data can be displayed.

1. In the Document window, select the bottom table row by positioning the pointer at the left edge of the table row until the pointer changes to a right arrow, and then clicking once.
2. In the Server Behaviors panel (Window > Server Behaviors), click the Plus (+) button and select Repeat Region.
3. In the dialog box (see Figure 11), click OK to accept the default settings.

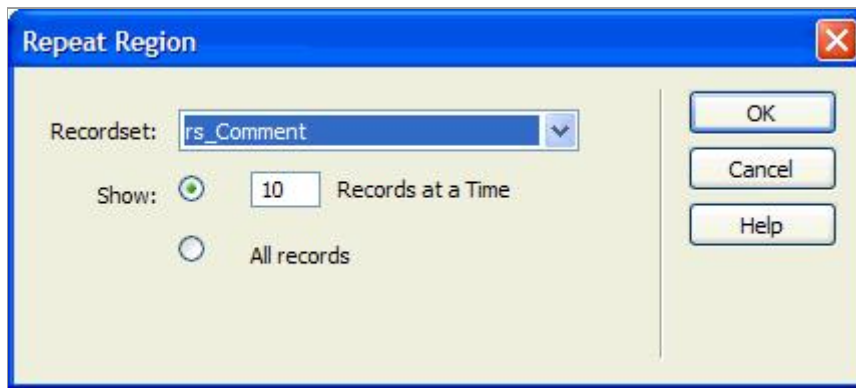


Figure 11. The Repeat Region dialog box.

An outline appears around the table row, and a tab appears above it and to the left to indicate that it's a repeated region. (The tab's label varies depending on the server technology you're using.)

4. Save your page.

Preview your page

Next, you'll preview the page to see how it looks with actual data in the table. To preview a page with data in Dreamweaver, you can use Live Data view.

With the file `view.php` (or `view.cfm`, or `view.asp`) still open in Dreamweaver, select **View > Live Data**.

The page is rendered with the recordset data (see Figure 12).

Customer Comments			
First Name	Last Name	E-mail	Comments
Dieter	Dietrich	dd@worldly.com	Is your London restaurant open on Saturdays?
Sachiko	Matsuda	sachi@web.net	Unbelievably excellent service received at your New Tokyo restaurant.
Ben	Morin	bmorin@seadec.com	I would like more seafood choices on the menu. Thanks.
Letitia	Riley	letitianiley@juno.com	Are there any plans to open a restaurant in Anchorage, Alaska?

Figure 12. Using Live Data to preview the page.

You can continue to design your page in Live Data view. For example, you can add a background color to the heading row, or right-align the text in the E-mail column. To do the latter, select the E-mail column, click the **Align Right** button on the Property inspector, and then click the **Refresh Live Data** icon on the left of the Live Data toolbar (or press **Control+R**).

You can also preview the page in a browser by selecting File > Preview in Browser.

Create a record insert form

The next page you'll create for the Cafe Townsend website enables customers to add comments to the database.

Dreamweaver includes several application objects that help you create web application pages quickly and easily. You'll use the Record Insertion Form application object to create the insert page. This object creates an HTML form, data fields that correspond to fields in the database, and the necessary server scripts to create a dynamic page.

Add a Record Insertion Form application object

You can use a Record Insertion Form application object to create a form that enables visitors to enter data into a database. The application object lets you select which fields to include in the form, label the fields, and select the types of form objects to insert. When a user enters data in the form fields and clicks the submit button, a new record is inserted in the database. You can also set a page to open after a record has successfully been submitted, so that the submitter knows that the database was updated.

1. In the Files panel, locate the file named *send* (send.php, send.cfm, or send.asp) and double-click the file to open it.
2. Place the insertion point in the empty table row below the row that contains the paragraph.

You want to insert the application object in the empty row.

3. Select Insert > Data Objects > Insert Record > Record Insertion Form Wizard. The Record Insertion Form dialog box appears.
4. In the Data Source (ColdFusion) or Connection (PHP and ASP) pop-up menu, select connTownsend. In the Table pop-up menu, make sure COMMENTS is selected.
5. Click the Browse button next to the After Inserting, Go To text box.
6. In the dialog box that appears, select the file view.php (or view.cfm, or view.asp) and click OK to close the dialog box.

For testing purposes, you want to see the page that displays all of the comments after you enter a comment. After testing the web application, change this setting to show a page that thanks the visitor for the comment.

Figure 13 shows what the dialog box should look like at this point.

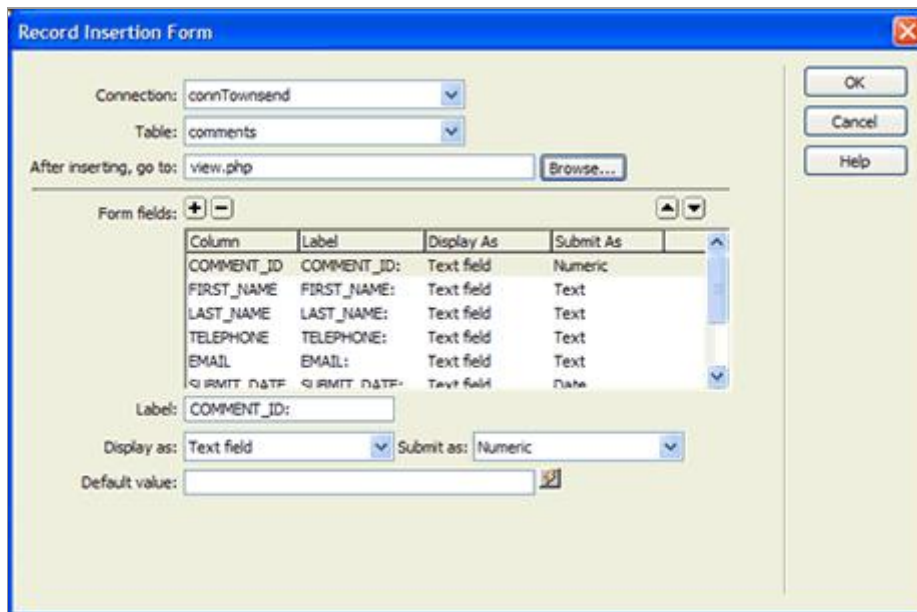


Figure 13. The Record Insertion Form dialog box.

7. Complete the rest of the wizard by following the procedure in the next section.

Create the insert form

In the Form Fields section of the Record Insertion Form Wizard, you define the form that a visitor enters data into.

1. In the Record Insertion Form Wizard, remove the fields you don't want included in the form by doing the following:
 - Select COMMENT_ID and click Minus (-).
 - Select TELEPHONE and click Minus (-).
 - Select SUBMIT_DATE and click Minus (-).
 - Select ANSWERED and click Minus (-).
2. In the Form Fields list, select FIRST_NAME to specify how the field will be displayed in the form.
3. In the Label text box, type **First Name** to replace the default text. This is the label that will appear in the HTML form next to the text box.
4. In the Display As pop-up menu, accept the default value Text Field; in the Submit As pop-up menu, accept the default value Text.
5. Leave the Default Value field empty.

For some web applications, you would use this field to set an initial text directive for the field to let the user know what kind of information to provide. In this case, the field label First Name makes clear that the visitor is supposed to enter his or her first

name.

When finished, the FIRST_NAME entry should look similar to the example in Figure 14.

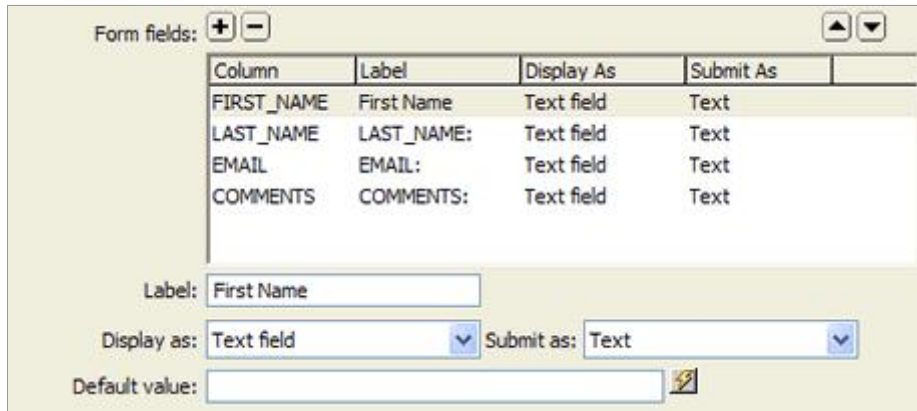


Figure 14. The FIRST_NAME field in the Form Fields section of the Record Insertion Form wizard.

6. Repeat Steps 3 through 6 for the LAST_NAME form field, with a label of Last Name, displaying as a Text Field and submitting as Text.
7. Repeat Steps 3 through 6 for the EMAIL form field, with a label of Email, displaying as a Text Field and submitting as Text.
8. In the Form Fields list, select COMMENTS.
9. In the Label field, enter **Comments**.
10. Select Text Area in the Display As pop-up menu, and accept the default value Text in the Submit As pop-up menu.

Figure 15 shows what the dialog box should look like when you are done.

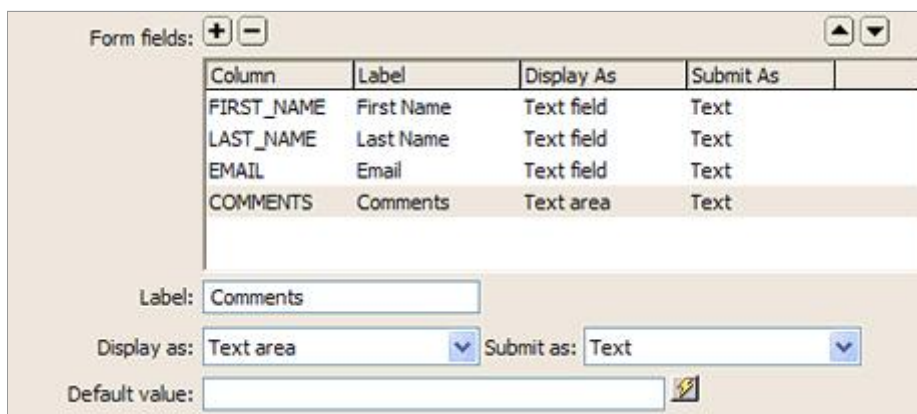


Figure 15. The COMMENTS field in the Form Fields section of the Record Insertion Form wizard.

11. Click OK to close the dialog box and create the record insertion form. The Record Insertion Form application object is inserted in the document. Figure 16 shows the completed customer feedback form.



Figure 16. Page displaying the completed customer feedback form.

12. Save your page.

Testing your application

The final step is to thoroughly test your application. In this section, you'll view the Insert Record page in a browser, add a test comment or question, and submit the data to test the application.

Note: If you're using a remote server (for example, if you're a Macintosh user working with ASP), you must upload the application files to your server. In the Files panel, select the site root folder and then click the blue Put Files arrow icon to upload the files to your server.

To test your application:

1. With the file named send.php (or send.cfm, or send.asp) as the active document, select File > Preview in Browser or press F12 to view your page in a browser.
2. Enter test data in the form and click Insert Record in the document to submit your data.

The application server receives the submitted information, updates the database, and displays the View page with the new information.

Note: If you're using the Microsoft Access sample database and get an error message, it's probably a folder permissions problem. See [Troubleshooting permissions problem](#) in the Dreamweaver LiveDocs.

Conclusion

This tutorial provided an introduction to a set of tools in Dreamweaver that enables you to create dynamic, data-driven pages. You learned how to create recordsets to define the data you want to use in your website. You also created a small web application that enables visitors to your site to interact with a database.

Where to go from here

For more detailed information on developing web applications with Dreamweaver, see the following topics in the Dreamweaver LiveDocs.

- [Data sources for web applications](#)
- [Making pages dynamic](#)
- [Building applications visually](#)

About the author

Charles Nadeau is a documentation manager at Adobe, where he leads the development of end-user content for Dreamweaver and acts as the Web Suite content architect. He also managed the documentation for Flex Builder 1 and 2, Fireworks 8, and the last few releases of Dreamweaver. Charles started at the company by writing the help for Dreamweaver UltraDev 1 and 4 as well as numerous articles and tutorials. Charles' interests include web design and development, motorcycling, and history.