http://www.interaktonline.com/

Tutorial

Build a Contact Form

Table of contents

Introduction

By following this tutorial you will learn how to use **MX Kollection 3** features to quickly build a page that allows your site visitors leave a message for a certain department of your company.

The web application will allow you to:

1. Build a page that lets site visitors write messages to one of your company's departments
2. Save those messages in a database
3. Send the messages through e-mail to the respective departments.

This tutorial contains two main steps:

1. Creating the form that will only store the messages in a database.
2. Implementing the send e-mail option for the messages.

If you have the **MX Kollection 3** bundle installed, then you have all the needed tools. Otherwise, the following separate products should be installed on your computer in order to complete the *Contact Form* tutorial:

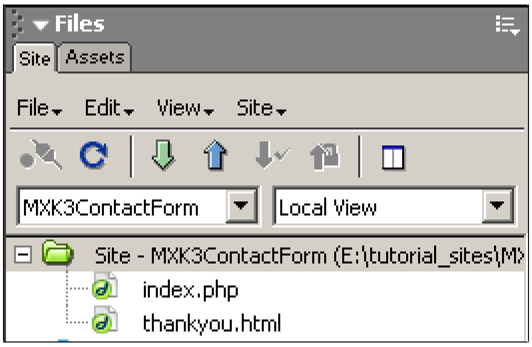
- **ImpAKT**
- **MX Send E-mail**
- **MX Form Validation** (optional - this product is used in the tutorial as well, but you can complete it without it also).

The estimated completion time for this tutorial is about 30 - 45 minutes. It depends on your authoring knowledge with **Macromedia Dreamweaver** (MX or MX 2004) and **MX Kollection 3**.

Plan the contact form

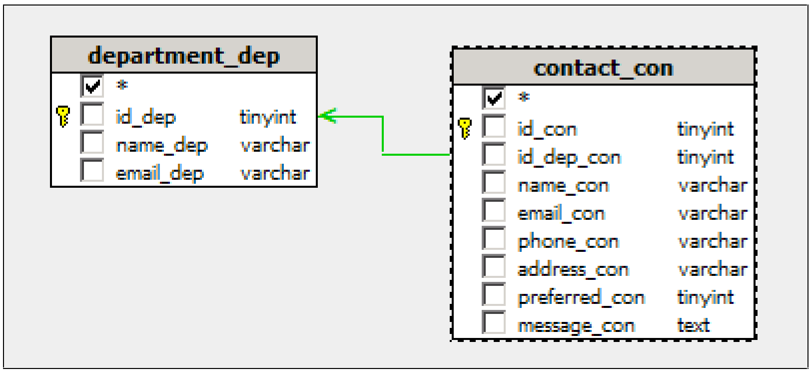
Before you start building this application, make sure you have a correctly configured Dreamweaver site, and a working database connection. For more instructions regarding these actions, consult the *Getting started* help file, which can be found in **Help -> InterAKT -> Getting Started**.

You will only have to create two files in your site's root. You can create them at the very beginning, so that you will not waste time with this operation again. To create files and folders in the site's root, use the corresponding options in the **File** menu of the **Files** tab. All files created in this tutorial can also be found inside the downloaded package, and you can use them to compare your work with what has been already done. The file structure will look as in the example below, and you can create it easily by unpacking the *zip* file corresponding to your server model from *\tutorials\Contact Form* in your site root:



The index of your site will display the contact form and perform most of the actions, while the *thankyou.html* page is only used to show a message to visitors that submit their messages.

After having created the files for your pages, it is time to set up the database to store the information for display. For this tutorial, you will use two tables: one for the departments, and one for the visitor's messages. The tables and fields are shown below:



Note: The database diagram in the image above was built with **MX Query Builder** (also referred as **QuB**) to better illustrate the database structure. You do not need to build it in order to complete this tutorial.

Here's a listing of the tables and columns used in this database:

1. *department_dep* - stores the departments information.
   - ***id_dep*** - the primary key for the department table. No two departments will have the same ID.
   - ***name_dep*** - the department name (Sales, IT).

  - ***email_dep*** - this field stores the email address of the department, (sales@interaktonline.com).

2. *contact_con* - the contact table stores a list of contents for each department. It is linked to the department table by the ***id_dep_con*** foreign key, which stores the ID of the department associated to each contact record.
    - ***id_con*** - the primary key for the contact table. No two contacts will have the same ID.
    - ***id_dep_con*** - the foreign key to the department table. This field stores the same value as ***id_dep***.
    - ***name_con*** - the name of the contact (Rob Anderson, Dan Smith).
    - ***email_con*** - the contact's email address (rob_anderson@somedomain.org).
    - ***phone_con*** - the contact's phone number.
    - ***address_con*** - address of each contact.
    - ***preferred_con*** - here the preferred means of contact is stored (phone, email, postal mail).
    - ***message_con*** - this field stores the content of the email message.

You can find the scripts needed to create an identical table structure inside the downloaded package, in the *\tutorials\Contact Form\db* folder, as an *sql* or *mdb* file, depending on the database server you intend to use. Import them in your database server management software (e.g. **PHPMyAdmin**, **Microsoft Access** etc.).

Once you have created the index of your site, open it, and if you haven't already, create a database connection to the tables you've just created. Throughout this tutorial, the connection used will be called *connForm.*

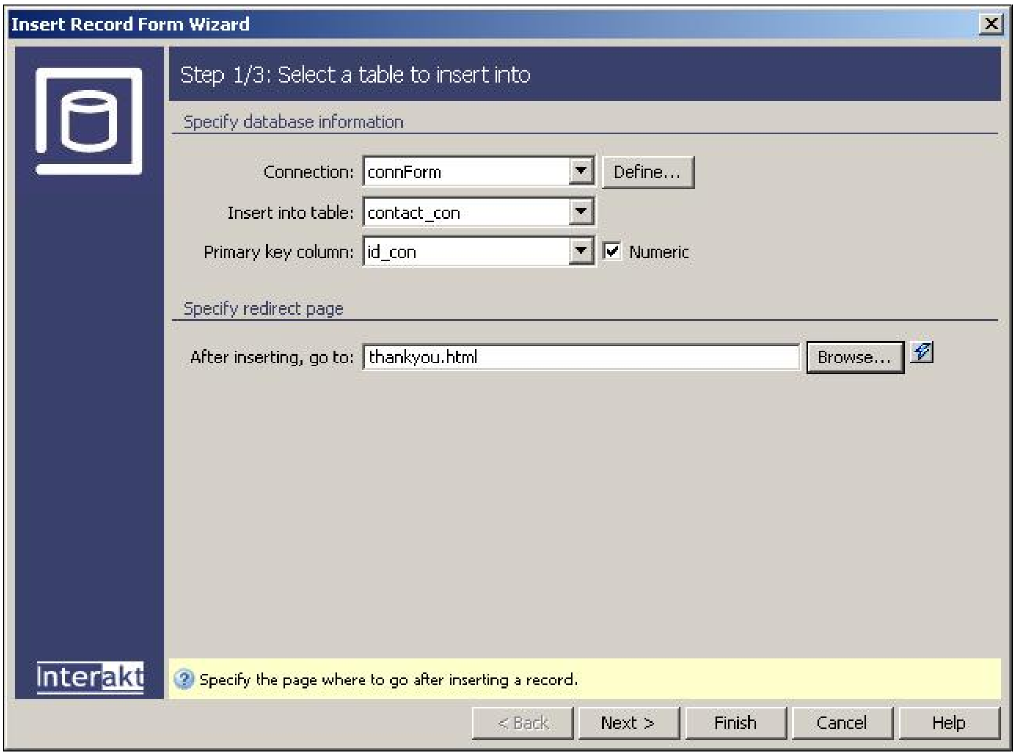
Create the contact form

In this section of the tutorial, you will use **ImpAKT** to build the contact form. It will save messages into the database, through the connection you've created when planning out the application.

1. Open the *index* page.
2. Start the **Insert Record Form Wizard**. You can access this wizard in two ways:
    - From the **MX Kollection** tab of the **Insert** panel, by pressing the first button.
    - From the **Server Behaviors** tab, by pressing the **Plus** (**+**) button, and then selecting **MX Kollection** -> **Forms** -> **Insert Record Wizard**.

The **Insert Record Form Wizard** user interface opens, allowing you to set options regarding the table, form elements and, if **MX Validation** is installed, rules for validating the user input. You can follow the steps below, or simply use the images as a guideline.

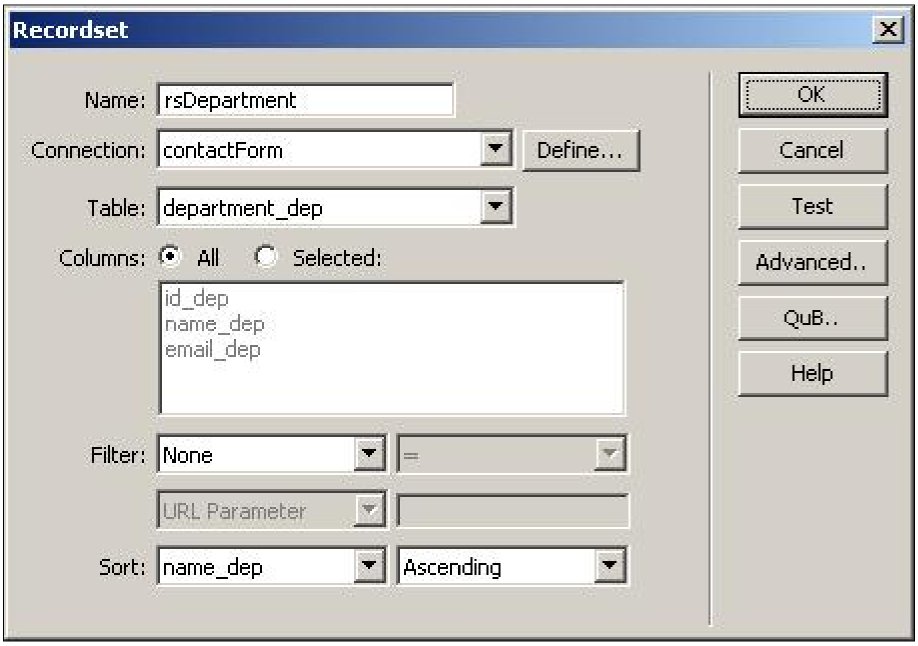
1. In the first step, you must define a database connection, the table where to store data into and where to go after the page is successfully submitted to the server.
    - In the ***Connection*** drop-down menu, select *connForm*.
    - In the ***Table*** drop-down menu, select *contact_con* (we don't want to add departments, but the contact form contents).
    - Leave the Primary key ***column*** at its default, the ***id_con*** field.
    - In the ***After inserting, go to*** text-box, you'll have to enter the *thankyou.html* file. Either type the file name, or use the **Browse** button to locate it.



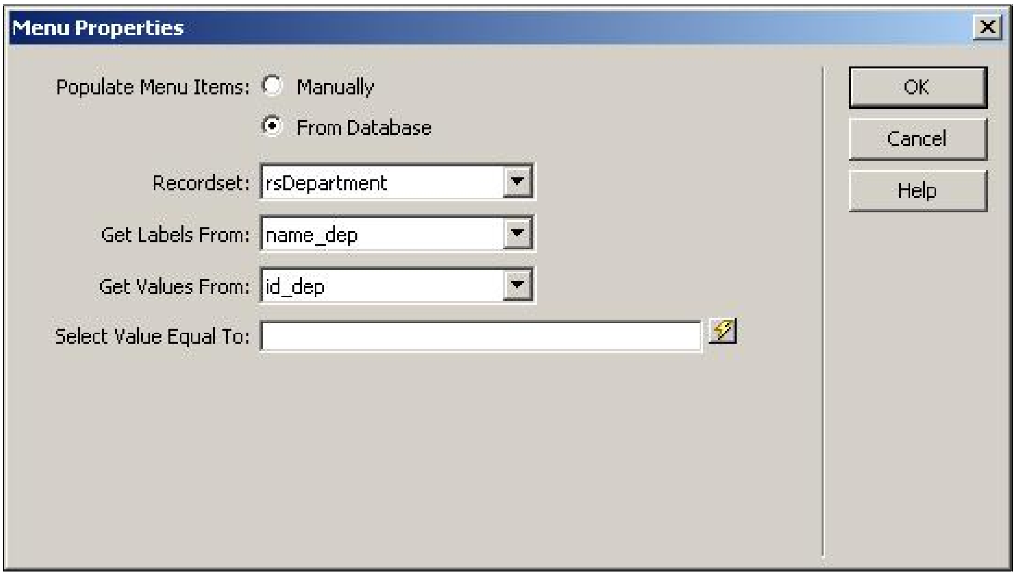
    - Once done with this step, click the **Next >** button to proceed to the field definitions.

2. In the second step, you can configure how each field will be represented in the page, how it will be submitted, and what labels they have. Most of the fields are set by default, with adjustments being made for the fields with fixed values, as the department name and the preferred method of contact:

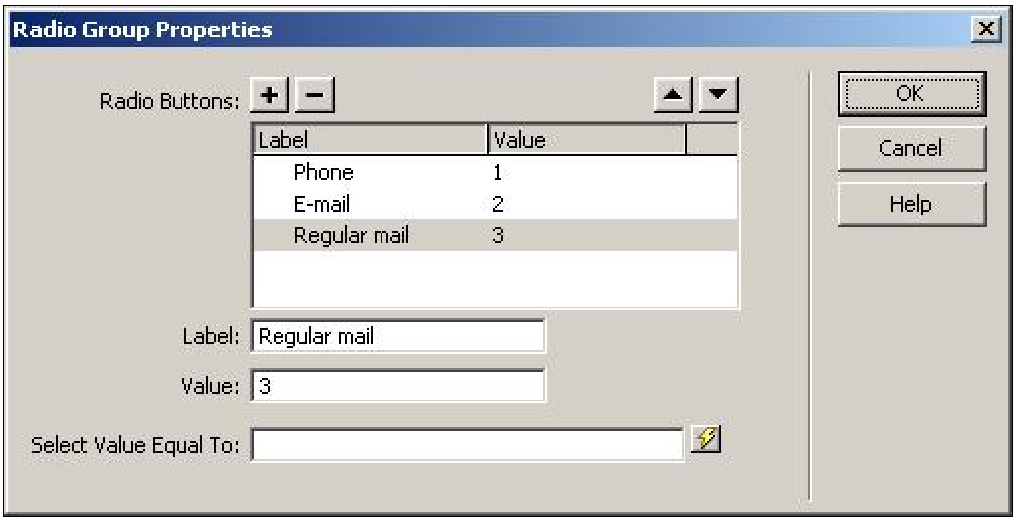
   - For the departments field, a drop-down menu will be used. To set it up, select the *id_dep_con* field in the grid, and set its ***Display as*** property to **Menu**. Upon selection, two new options will become available: the **Menu Properties** and **Add Recordset** buttons.
   - Since the department information is stored in a database table, you have to create a recordset to retrieve it. Just hit the **Add Recordset** button, and use it to retrieve all of the departments from the *department_dep* table:



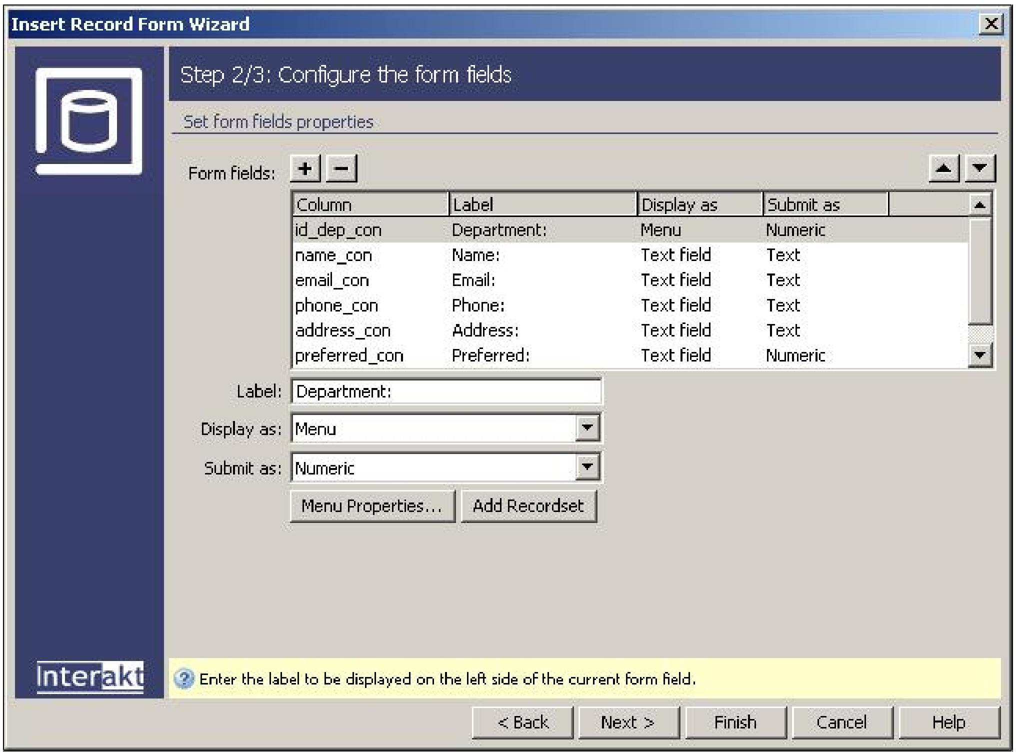
   - Once you've set up the recordset, hit the **OK** button. Now you can associate its values to the menu. To do so, click on the **Menu Properties** button and set it to retrieve the menu values ***From Database***:



- The department table fields containing the labels and values are the department name (**name_dep**) and the department unique identifier (**id_dep**). Once configured, click the **OK** button to finish.
- The second field to change the preferred method of contact, which lets your company communicate through the means they like most: through e-mail, phone or regular mail. To show these options in a easy-to-understand manner, you will use radio buttons. Select the *preferred_con* field in the grid, and set its ***Display as*** property to **Radio group**. Once you do this, a new option will become available: the **Radio Group Properties** button.
- Once you click on the **Radio Group Properties** button, you open a dialog box where the labels and values for each radio button in the group can be set. Enter the "Phone", "E-mail" and "Regular mail" as labels, and 1,2 and 3 as values. Hit **OK** to return to the wizard:

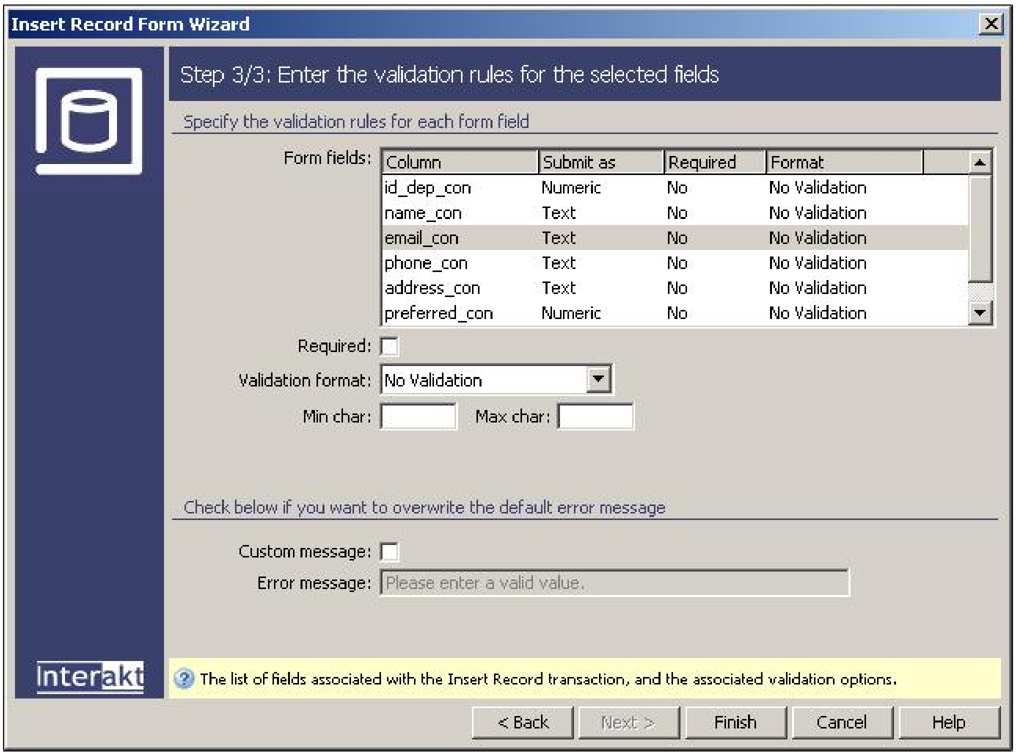


- The last field to set is the message. Set it to be displayed as a **Text area**, instead of a **Text field**, to allow visitors enter more text. You can then move on to the last (optional) step of the wizard: validation rules.



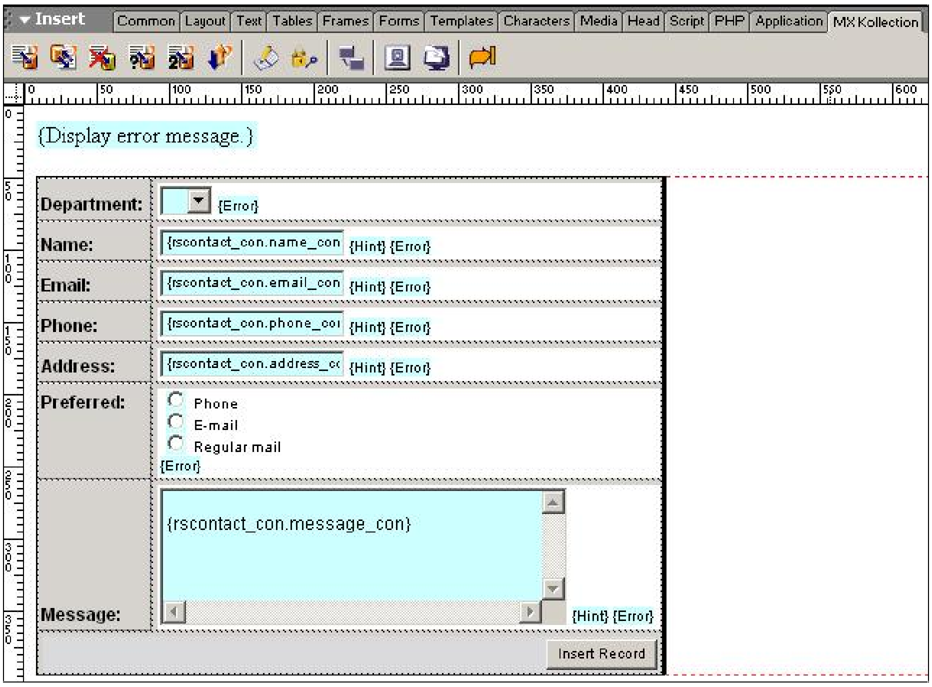
3. In this last step you can add a validation rule for each field. If the user enters bad data, it will be processed first in the browser, and an error message will be displayed. Since you have to somehow identify the user, you should add validation rules and required status for the name, e-mail address and message, in the following manner:

   - For the name and message field, you only have to set them as required. To do so, select each of them in the grid, and check the **Required** checkbox.
   - For the e-mail address field, aside checking the ***Required*** checkbox, select the ***E-mail*** option in the ***Validation format*** drop-down menu:

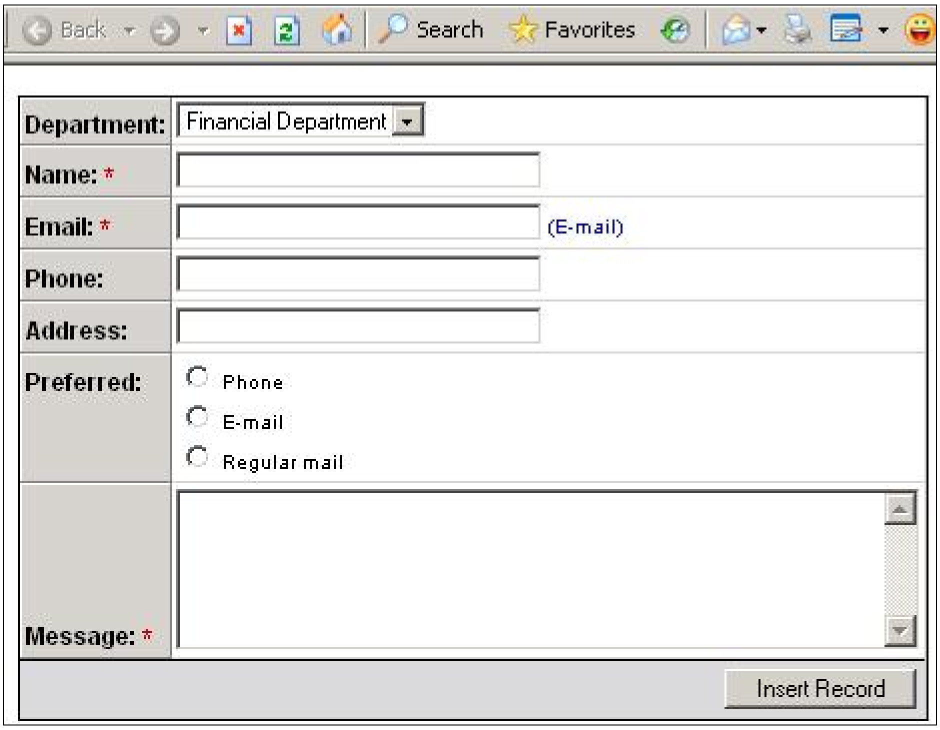


4. With this last step, the contact form is built. You can now click on the **Finish** button to apply the wizard to the page.

You will notice all of the elements it adds to the page: all of the **HTML** form elements, the recordset created for the menu, the **Form Validation** triggers, and the **Insert Record** transaction. The finished form should look like the image below (in the **Dreamweaver**'s editable area):

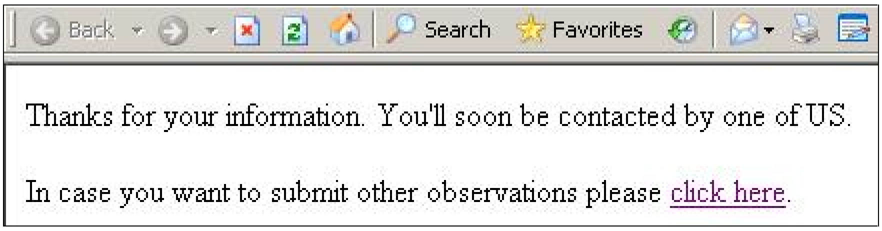


Above the actual form, a translator indicates the place where the main error message will be displayed if the transaction fails. After you save the page, you can also test it, by loading it in your browser and submitting a message.



Note: the red * showing that the field is required, as well as the "*(E-mail)*" mark, showing what kind of data you must enter.

Before moving continuing with the page creation, there is one more thing left to do in order for the contact form to be usable: create the `thankyou.html` page. Open it in **Dreamweaver**, and type in an appreciation message for their opinion. Also, you should add a link to the `index` page, for those who still have things to say. An example page is in the tutorial folder:



Select e-mail recipient

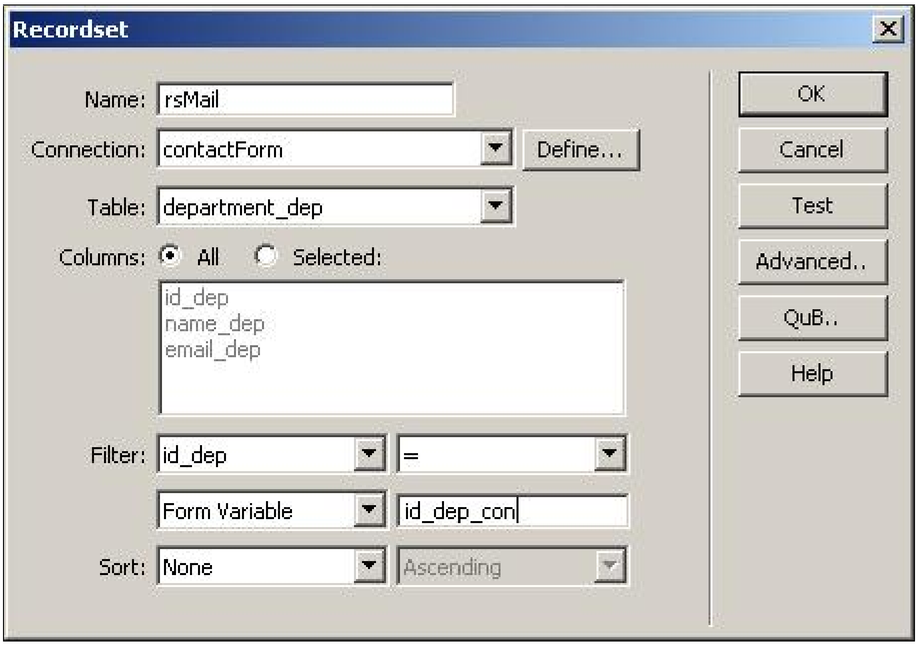
Next you will add the capability to send the message through e-mail.

There are two steps involved in the implementation:

- Retrieve the correct e-mail address right after the page is submitted.
- Add the send e-mail operation, through the **Send E-mail** trigger .

Open the `index` page. To retrieve the correct address, you will have to add a filtered recordset into the page. To correctly configure it, follow the instructions below:

1. Click on the **Plus** (**+**) button of the **Bindings** tab. If the **Bindings** tab is not visible, open it from **Window** -> **Bindings**.
2. In the menu that appears, select ***Recordset (Query)***. The new recordset dialog box will then open.
3. Give the recordset a significant name (e.g. *rsMail*)
4. Select your database connection and the `department_dep` table.
5. At this point the recordset grabs all departments, so you will add a filter to just grab the one you need :
    - In the ***Filter*** drop-down menu select the department's unique identifier, which must be equal to a ***Form Variable***. The form variable's name must be written in the text field. In this case, it is ***id_dep_con***. Either type it, or paste it in the text-box.

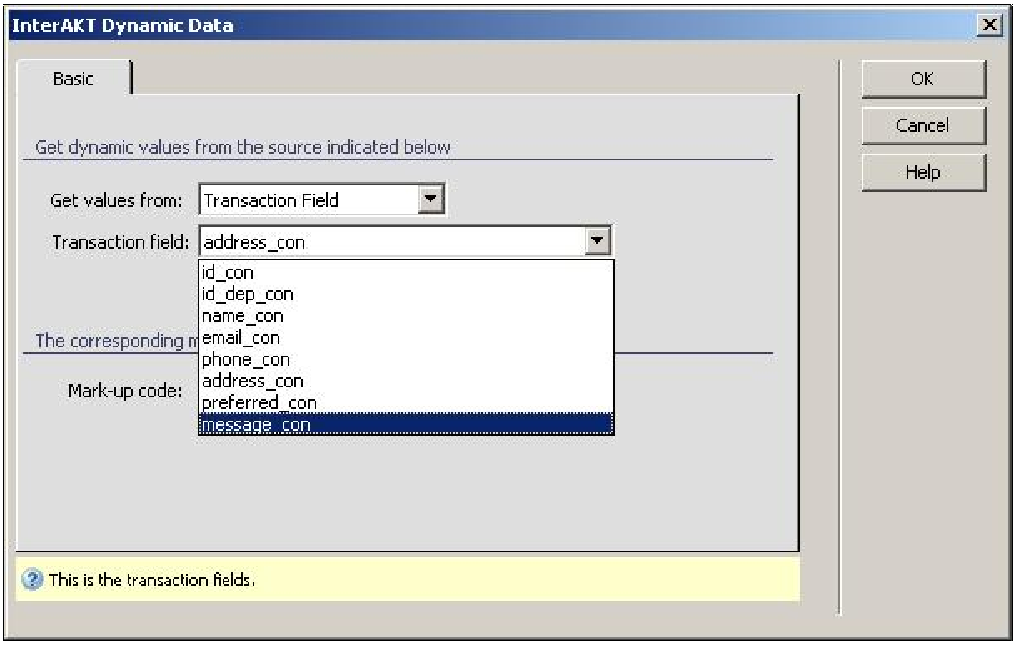


    - Now, the recordset only retrieves one department: the one selected in the form.
    - Click the recordset dialog box's **OK** button to apply the changes.

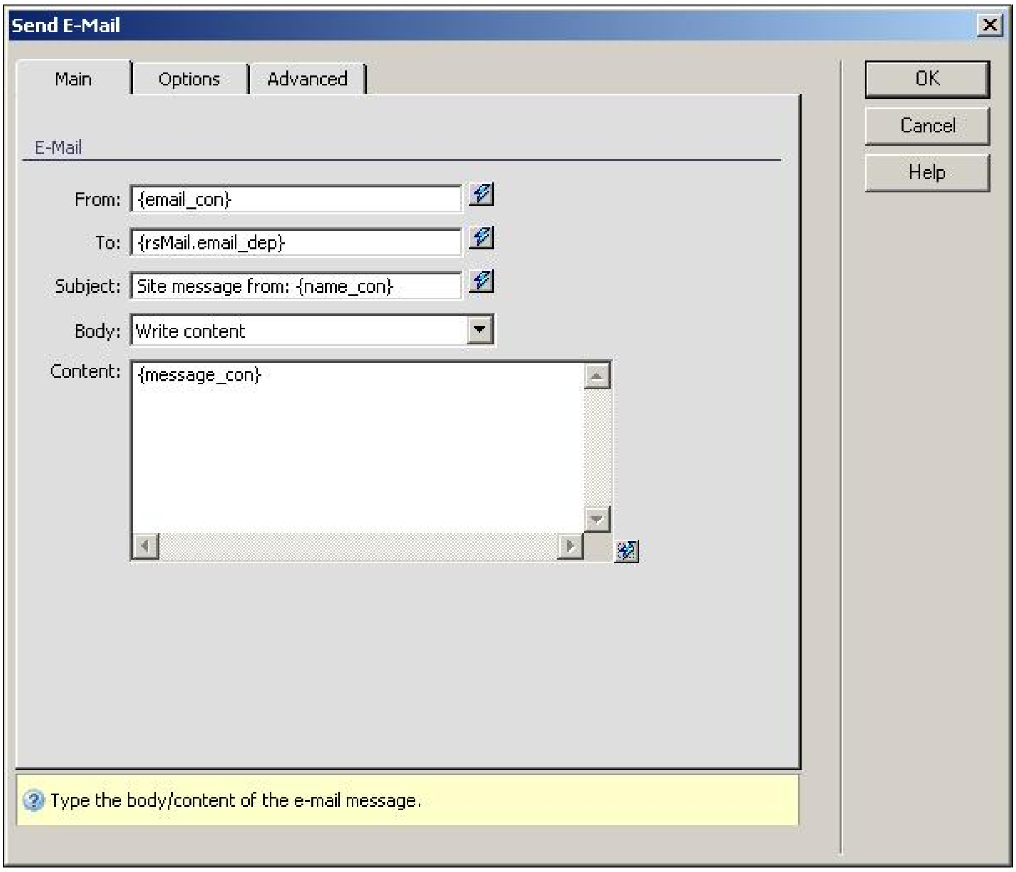
For the second part of the enhancement, the e-mail sending operation, you will use the **Send E-mail** trigger. Follow the next steps to add it:

1. Open the Send E-mail trigger dialog box. You can access it from the Servers Behaviors tab, by clicking on the Plus (+) button, and selecting MX Kollection -> Send E-mail -> Send e-mail.
2. To set the mail trigger options, you will use the InterAKT Dynamic Data. This is a way to add dynamic placeholders that get replaced at runtime with their correct values.

3. To access the Dynamic Data dialog box, click the blue lightning box next to each text field. Except for the To field, all other dynamic data is retrieved from the transaction fields:



4. For the **To** dialog box field, the dynamic value is retrieved from the filtered recordset created at the previous step.
5. The trigger's completed first tab should look like in the following image:

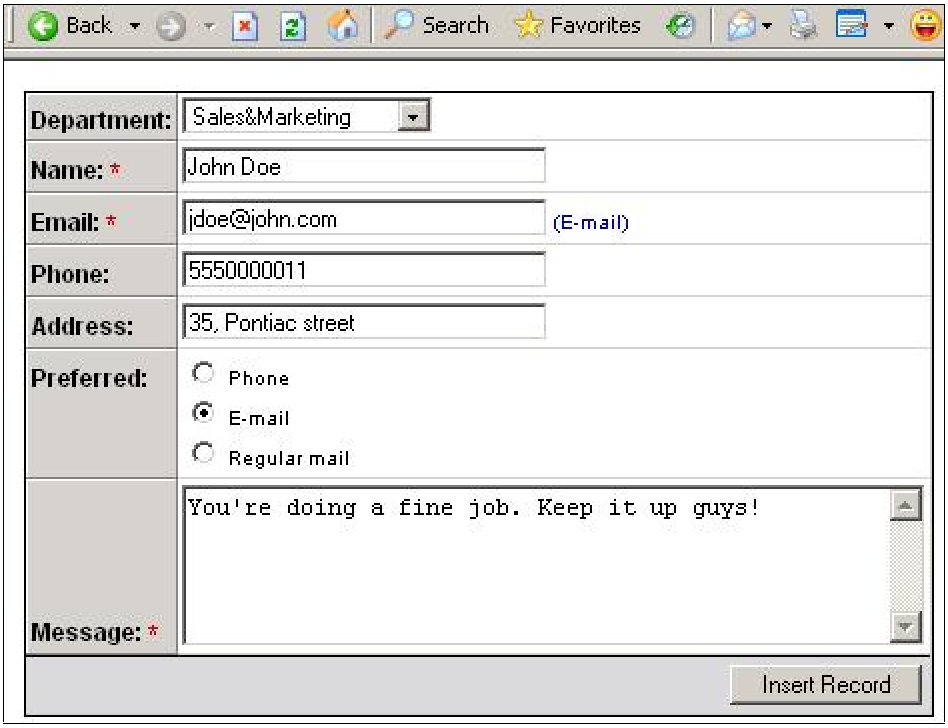


6. This is the only step that needs to be completed for the purpose of this tutorial. Click the **OK** button to apply the **Send e-mail** trigger to the page.

Note: If using the **ASP VBScript** server model you must configure the E-mail server settings in the Control Panel. You must fill in the server address (or name), the port (by default it is 25), user name and password. Optionally, you can also set the default sender field.

Save the page and preview it in the browser. Fill in a message for a department, and submit the page, similar to what you've done as a test in the previous section. Only that now you just need to check that department's e-mail to read your message.

The message in the contact form, as seen in the browser:



The mail message delivered to the department:

At this point the contact form is fully functional, and it allows sending the messages through e-mail, and also keeping them in the database.

Other Resources

Other Dreamweaver Extensions from InterAKT

- KTML
- MX Kart
- MX Site Search
- MX RSS Reader-Writer
- MX Dynamic Table Sorter
- MX Coder Pack
- MX Dynamic Charts
- MX CSS Dynamic Menus

Contact Us

- **Address:**
  1-11 Economu Cezarescu ST, AYASH Center, 1st floor
  Sector 6, ZIP 060754, Bucharest, Romania
- **Web:** http://www.interaktonline.com/
- **E-mail:** contact@interaktonline.com
- **Phone:** +4031 401.68.19 or +4021 312.51.91
- **Fax:** +4021 312.53.12

Copyright

Send comments and suggestions to products@interaktonline.com

InterAKT Online

Web: http://www.interaktonline.com/
E-mail: contact@interaktonline.com
Address: 1-11 Economu Cezarescu ST, AYASH Center, 1st floor, Sector 6, ZIP 060754, Bucharest, Romania
Phone: +4021 312.51.91
Fax: +4021 312.53.12