

---

## In This Part

---

Tip 1: Creating Tables for Web Applications: An Introduction to Calculated Fields

Tip 2: Alternative to the Hyperlink Data Type

Tip 3: Fill a Table with Numbers

Access is a relational database management system (RDBMS), so it's critical that the tables be properly designed for any application to be successful. We felt that it wasn't appropriate to provide table-related tips in isolation, but there were a few items that we thought were worth including.

Access 2010 introduces a new concept of *calculated table fields* in order to be consistent with SharePoint. Note that the use of calculated fields is not limited to use in Access web applications. However, be aware that some of the new field types, including the calculated, attachment, and lookup fields, are not traditionally used in classical relational design. As such, they may not scale well to other systems. Even the traditional field type, hyperlink, has alternatives worth considering, as you will see in Tip 2.



# Creating Tables for Web Applications

## *An Introduction to Calculated Fields*

### Objective

---

This Tip introduces you to designing tables for Web-based applications.

Tables in Access 2010 have a new feature called *calculated fields* that allows expressions (*calculations*) to be stored in a table. Whereas this new feature is available for both Web and client applications, it has greater importance in Web-based applications. This is due to the importance of moving program and application logic out of forms and into the back-end Web server.

When you design a table for Web Services in Access, you do not have the traditional Design View for Access tables. When building a Web-based table, you are restricted to using this Table Layout View, which was introduced in Access 2007.

Web-based forms have substantially fewer program features than the standard client VBA-enabled forms that traditional developers have become so accustomed to over the years. Thus, you need to use shortcuts like calculated fields as a means to add additional functionality to your Web-based applications.

### Scenario

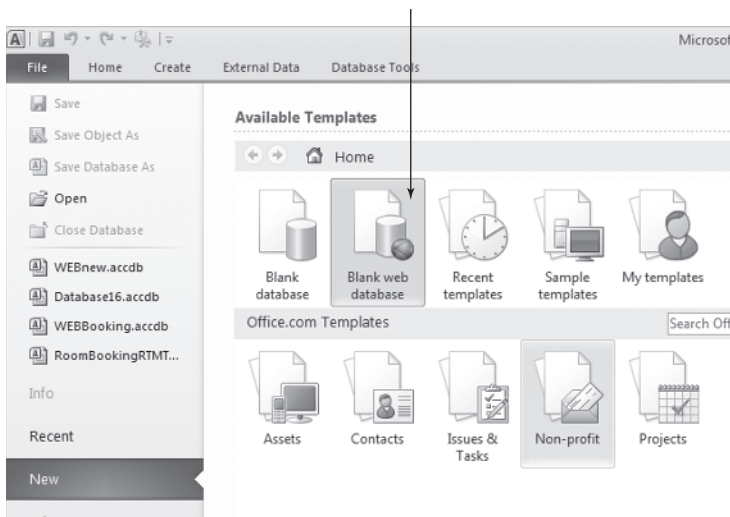
---

In this Tip, you learn how to build a table that allows you to enter a City value into a table and use a calculated field to store a resulting URL that points to a weather website for the given city.

For longtime Access developers, having calculated expressions occur at the table level is a relatively new concept. Crossing the bridge into Web-based development results in forms that run on the user's desktop inside a Web browser. This means that code you write in a form also runs on the user's desktop. It follows that code running on the desktop inside a Web browser has limited ability to manipulate data directly in the table that is now sitting on the Web server. For Web applications, this is a significant change for developers, who by nature stuff as much code and logic as possible into Access forms. The new goal in building Web applications is to move program logic, expression logic, and calculation logic out of the form running on the user's desktop and into the data tables, which are located on the Web server.

## Tables

To create Web-based forms and tables, you must first create a Web database, as shown in Figure 1-1.



**Figure 1-1:** You must choose a Web database to create Web-based applications.

**NOTE** The file extension for Web databases is the same `accdb` extension as for regular databases. However, the Web file format is different from that for regular Access databases. For this reason, it's a good idea to come up with some type of naming convention to distinguish between regular Access databases and Web databases.

The goal here is to build a simple list of cities and store a weather URL for each city. (This URL is used in Tip 36 to feed the Web Browser control to display the current outside temperature and weather for a given city.) This table can then be used to

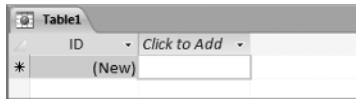
enable your Access forms to display the current weather and temperature for a given city location.

The simple and basic design of our table is shown in Table 1-1.

**Table 1-1:** Table Design

FIELD NAME	TYPE	DESCRIPTION
ID	AutoNumber	(Primary Key)
City	Text (50)	City
Comments	Text (50)	Some general comments or notes
Units	Text (1)	Temperature units (degrees Celsius, C, or Fahrenheit, F)
MyURL	Text (255)	(Calculated Field) URL of website that displays the current weather

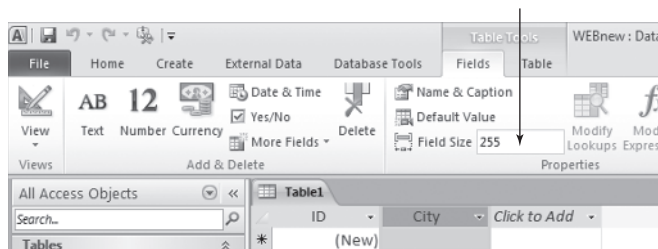
When you choose “Create Table,” keep in mind you don’t have the Table Design View, but are restricted to Layout View (shown in Figure 1-2).



**Figure 1-2:** Table design is restricted to Layout View.

Choose “Click to Add” to add a field to the table. Note how the ID field is already created. As a general tip, it is strongly suggested that you do not change the data type or name of the primary key field when designing tables for Web-based applications. (Note that Web-based applications do not support compound primary keys.) If you don’t have a primary key field when you upload your table into Access Web Services, then an ID field is created for you.

For the first field (*City*), you simply choose `Text` from the dropdown list of field types, and type in `City` for the name of the field. Whereas the default width for `Text` fields is 50 characters in client applications, it’s 255 characters in Web-based applications. To set the length of Web-based tables, use the Fields tab in the Ribbon, as illustrated in Figure 1-3.



**Figure 1-3:** Use the Ribbon to set the field size.

Add the *Comments* and *Units* fields in the same manner.

To add *MyURL*, the calculated field to hold the weather URL, select “Calculated Field” from the dropdown list of field types, then *Text* as the type of calculated field, as shown in Figure 1-4. You will then see the standard Access Expression Builder dialog (see Figure 1-5).

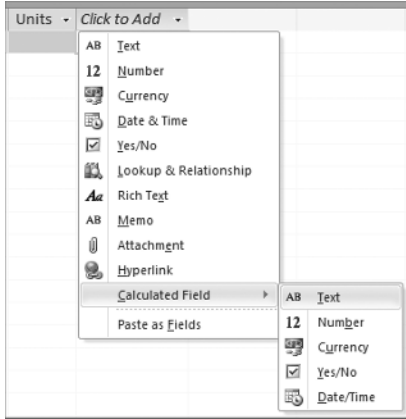


Figure 1-4: Creating a calculated field.

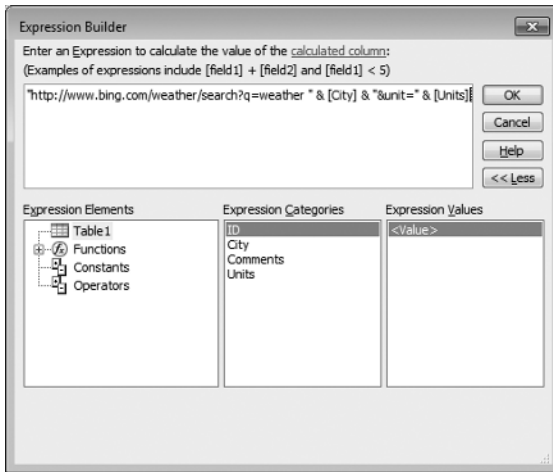


Figure 1-5: Using the Expression Builder to create a calculated field.

The URL we use here is based on Bing’s weather search, defined as `www.bing.com/weather/search?q=weather City&unit=C`.

This means that we need to create a regular Access expression that adds the values for *City* (the city name) and *Units* (C or F, for Celsius or Fahrenheit) to the base URL above. As is shown in Figure 1-5, this is done using the following expression:

```
"http://www.bing.com/weather/search?q=weather " & [City] & "&unit=" & [Units]
```

Note how the above code is made up of a string that includes two fields from the table. Any update to either field causes Access to recalculate this expression for you and *then* save the results as a field. This expression is thus maintained at the data engine (table) level by Access. It is possible, by the way, to use the result of the expression *before* the record is saved to disk, should you need to.

Save your table as *tblCityWeather*, and you're done.

Try entering a record in Table View and watch the calculated field update for you! See Figure 1-6.



ID	City	Comments	Units	MyURL
1	New York		C	http://www.bing.com/weather/search?q=weather New York&unit=C
* (New)				

**Figure 1-6:** The value for the MyURL field is calculated for you.

---

## (Web) Queries

No queries are used in this Tip.

---

## (Web) Forms

No forms are used in this Tip.

---

## (Web) Reports

No reports are used in this Tip.

---

## (Web) Macros

No macros are used in this Tip.

---

## Modules

No modules are used in this Tip. (In actual fact, it is not possible to create modules in Web databases.)

## Using This Tip

---

Realistically, there is nothing that can be imported from the sample database. Instead, this Tip illustrates the concepts you must use to build your own tables for Web applications.

## Additional Information

---

Tip 36 illustrates how the value in the *MyURL* field added to the table can be used in a form that allows you to enter the city and then displays the current weather for that city.

This Tip is provided by Access MVP Albert Kallal. For more tips from Albert, check his website at [www.members.shaw.ca/AlbertKallal/index.html](http://www.members.shaw.ca/AlbertKallal/index.html).