## Access 2010: Creating Forms

### Table of Contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INTRODUCTION TO FORMS</strong></td>
<td>4</td>
</tr>
<tr>
<td>Create a form by using the Form tool</td>
<td>4</td>
</tr>
<tr>
<td>Create a split form by using the Split Form tool</td>
<td>4</td>
</tr>
<tr>
<td>Create a form that displays multiple records by using the Multiple Items tool</td>
<td>4</td>
</tr>
<tr>
<td>Create a form by using the Form Wizard</td>
<td>5</td>
</tr>
<tr>
<td>Create a form by using the Blank Form tool</td>
<td>5</td>
</tr>
<tr>
<td><strong>UNDERSTAND LAYOUT VIEW AND DESIGN VIEW</strong></td>
<td>5</td>
</tr>
<tr>
<td>Layout view</td>
<td>5</td>
</tr>
<tr>
<td>Design view</td>
<td>5</td>
</tr>
<tr>
<td>Modify a form in Layout view</td>
<td>5</td>
</tr>
<tr>
<td>Modify a form in Design view</td>
<td>6</td>
</tr>
<tr>
<td><strong>UNDERSTAND LAYOUTS</strong></td>
<td>6</td>
</tr>
<tr>
<td>Create a new layout</td>
<td>7</td>
</tr>
<tr>
<td>Switch a layout from Tabular to Stacked, or from Stacked to Tabular</td>
<td>7</td>
</tr>
<tr>
<td>Add rows or columns to a layout</td>
<td>8</td>
</tr>
<tr>
<td>Remove rows or columns from a layout</td>
<td>8</td>
</tr>
<tr>
<td>Rearrange controls in a layout</td>
<td>8</td>
</tr>
<tr>
<td>Add controls to a layout</td>
<td>8</td>
</tr>
<tr>
<td>Add a new field from the Field List pane to an existing control layout</td>
<td>8</td>
</tr>
<tr>
<td>Remove controls to an existing control layout</td>
<td>8</td>
</tr>
<tr>
<td>Remove controls from a layout</td>
<td>9</td>
</tr>
<tr>
<td><strong>INTRODUCTION TO CONTROLS</strong></td>
<td>9</td>
</tr>
<tr>
<td>Add a field to a form</td>
<td>10</td>
</tr>
<tr>
<td>Add a field to a form by using the Field List</td>
<td>10</td>
</tr>
<tr>
<td>Learn more about the Field List</td>
<td>12</td>
</tr>
<tr>
<td>Add a field to a form without using the Field List</td>
<td>13</td>
</tr>
<tr>
<td>Create a form by using the Form tool</td>
<td>14</td>
</tr>
<tr>
<td>Create a single item form</td>
<td>14</td>
</tr>
<tr>
<td>Create a form by using the Multiple Items tool</td>
<td>15</td>
</tr>
<tr>
<td>Create a multiple item form</td>
<td>15</td>
</tr>
<tr>
<td>Create a form by using the Form Wizard</td>
<td>15</td>
</tr>
<tr>
<td>Start the Form Wizard</td>
<td>15</td>
</tr>
<tr>
<td>Create a form by using the Blank Form tool</td>
<td>16</td>
</tr>
</tbody>
</table>

Topics came directly from Access 2010 Help.
<table>
<thead>
<tr>
<th>CREATE A TABBED FORM</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE A NAVIGATION FORM</td>
<td>18</td>
</tr>
<tr>
<td>CREATE A SPLIT FORM</td>
<td>19</td>
</tr>
<tr>
<td>CREATE A FORM THAT CONTAINS A SUBFORM</td>
<td>22</td>
</tr>
<tr>
<td>SET THE TAB ORDER FOR CONTROLS</td>
<td>27</td>
</tr>
<tr>
<td>SET THE DEFAULT FORM THAT APPEARS WHEN YOU OPEN A DATABASE</td>
<td>28</td>
</tr>
<tr>
<td>CREATE A USER INTERFACE (UI) MACRO</td>
<td>28</td>
</tr>
</tbody>
</table>

CREATE A TABBED FORM

- ADD A TAB CONTROL TO A FORM ............................................. 17
- MOVE EXISTING CONTROLS TO A TAB PAGE ............................... 17
- DRAG FIELDS FROM THE FIELD LIST TASK PANNE TO A TAB PAGE ........................................................................ 17
- ADD OTHER CONTROLS TO A TAB PAGE ....................................... 17
- ADD A NEW TAB PAGE .................................................................. 18
- REORDER TAB PAGES .................................................................... 18
- RENAME A TAB PAGE .................................................................... 18
- DELETE A TAB PAGE ...................................................................... 18

CREATE A NAVIGATION FORM

- CREATE A NAVIGATION FORM ....................................................... 18
- ADD A FORM TO A NAVIGATION FORM ....................................... 18
- EDIT THE LABEL AT THE TOP OF THE NAVIGATION FORM .......... 19
- EDIT THE FORM CAPTION ........................................................ 19
- SET THE NAVIGATION FORM AS THE DEFAULT DISPLAY FORM .... 19

CREATE A SPLIT FORM

- CREATE A NEW SPLIT FORM BY USING THE SPLIT FORM TOOL ..... 20
- TURN YOUR EXISTING FORM INTO A SPLIT FORM ....................... 20
- FINE-TUNE YOUR SPLIT FORM .................................................. 20
  - Set the split form properties .................................................. 20
  - Fix the form separator in place ............................................... 21
  - Add a field to a split form ..................................................... 21
  - Delete a field from a split form ............................................. 21

CREATE A FORM THAT CONTAINS A SUBFORM

- OVERVIEW OF SUBFORMS ......................................................... 22
- CREATE OR ADD A SUBFORM ...................................................... 24
  - Create a form that contains a subform by using the Form Wizard .......................................................... 24
  - Add one or more subforms to an existing form by using the SubForm Wizard ............................................ 25
  - Create a subform by dragging one form onto another .......... 25
- CHANGE THE DEFAULT VIEW OF A SUBFORM ............................. 26
- ADD RELATED DATA TO A FORM WITHOUT CREATING A SUBFORM .............................................................. 27

SET THE TAB ORDER FOR CONTROLS

- CHANGE THE TAB ORDER FOR CONTROLS ................................ 27
- REMOVE A CONTROL FROM THE TAB ORDER ............................ 27
- CHANGE THE TAB BEHAVIOR FOR THE LAST FIELD IN THE TAB ORDER ............................................................ 28

SET THE DEFAULT FORM THAT APPEARS WHEN YOU OPEN A DATABASE

CREATE A USER INTERFACE (UI) MACRO

- OVERVIEW ................................................................................. 29
- WHAT’S NEW .............................................................................. 29
- CREATE A STANDALONE MACRO .................................................. 29
- CREATE AN EMBEDDED MACRO ................................................ 30
ADD ACTIONS TO A MACRO ........................................................................................................... 30
Step 1: Browse or search for a macro action................................................................................. 30
Step 2: Add an action to a macro .................................................................................................. 31
Step 3: Fill in arguments ............................................................................................................... 32
MOVE AN ACTION .......................................................................................................................... 32
DELETE AN ACTION ...................................................................................................................... 32
RUN A MACRO .................................................................................................................................. 32

USE EXPRESSIONS IN FORM AND REPORT CONTROLS ............................................................... 33
ENTER AN EXPRESSION IN A TEXT BOX CONTROL ...................................................................................... 33
GUIDE TO EXPRESSION SYNTAX .................................................................................................... 33
  Introduction ...................................................................................................................................... 34
OBJECTS, COLLECTIONS, AND PROPERTIES ..................................................................................... 35
IDENTIFIERS ....................................................................................................................................... 35
FUNCTIONS, OPERATORS, AND CONSTANTS ...................................................................................... 36
  Functions ......................................................................................................................................... 36
  Operators ....................................................................................................................................... 36
    Arithmetic operators ..................................................................................................................... 37
    Comparison operators ................................................................................................................... 37
    Logical operators .......................................................................................................................... 38
    Concatenation operators .............................................................................................................. 38
    Special operators .......................................................................................................................... 38
  Constants ........................................................................................................................................ 38
EXAMPLES OF EXPRESSIONS ........................................................................................................... 39
Introduction to forms

A form is a database object that you can use to create a user interface for a database application. A "bound" form is one that is directly connected to a data source such as a table or query, and can be used to enter, edit, or display data from that data source. Alternatively, you can create an "unbound" form that does not link directly to a data source, but which still contains command buttons, labels, or other controls that you need to operate your application.

This handout focuses primarily on bound forms. You can use bound forms to control access to data, such as which fields or rows of data are displayed. For example, certain users might need to see only several fields in a table with many fields. Providing those users with a form that contains only those fields makes it easier for them to use the database. You can also add command buttons and other features to a form to automate frequently performed actions.

Think of bound forms as windows through which people see and reach your database. An effective form speeds the use of your database, because people don't have to search for what they need. A visually attractive form makes working with the database more pleasant and more efficient, and it can also help prevent incorrect data from being entered.

Create a form by using the Form tool

You can use the Form tool to create a form with a single mouse-click. When you use this tool, all the fields from the underlying data source are placed on the form. You can start using the new form immediately, or you can modify it in Layout view or Design view to better suit your needs.

Create a split form by using the Split Form tool

A split form gives you two views of the data at the same time — a Form view and a Datasheet view.

A split form differs from a form/subform combination in that the two views are connected to the same data source and are synchronized with one another at all times. Selecting a field in one part of the form selects the same field in the other part of the form. You can add, edit, or delete data from either part (as long as the record source is updatable, and you have not configured the form to prevent these actions).

Working with split forms gives you the benefits of both kinds of forms in a single form. For example, you can use the datasheet portion of the form to quickly locate a record, and then use the form portion to view or edit the record.

Create a form that displays multiple records by using the Multiple Items tool

When you create a form by using the Form tool, the form that Access creates displays a single record at a time. If you want a form that displays multiple records but is more customizable than a datasheet, you can use the Multiple Items tool.

When you use the Multiple Items tool, the form that Access creates resembles a datasheet. The data is arranged in rows and columns, and you see more than one record at a time. However, a Multiple Items form gives you more customization options than a datasheet, such as the ability to add graphical elements, buttons, and other controls.
Create a form by using the Form Wizard

To be more selective about what fields appear on your form, you can use the Form Wizard instead of the various form-building tools previously mentioned. You can also define how the data is grouped and sorted, and you can use fields from more than one table or query, as long as you specified the relationships between the tables and queries beforehand.

Create a form by using the Blank Form tool

If the wizard or the form-building tools don't meet your needs, you can use the Blank Form tool to build a form. This can be a very quick way to build a form, especially if you plan to put only a few fields on your form.

Understand Layout view and Design view

Layout view and Design view are the two views in which you can make design changes to forms. You can use either view to perform many of the same design and layout tasks, but certain tasks are easier to perform in one view than they are in the other. This section describes the similarities and differences between Layout view and Design view, and shows you how to perform some common form design tasks in each view.

Layout view: Layout view is a more visually-oriented view than Design view. In Layout view, the form is actually running. Therefore, you can see your data much as it will appear when you are using the form. However, you can also change the form design in this view. Because you can see the data while you are modifying the form, this is a very useful view for setting the size of controls or performing almost any other task that affects the appearance and usability of the form.

Certain tasks cannot be performed in Layout view and require switching to Design view. In certain situations, Access displays a message telling you that you must switch to Design view to make a particular change.

Design view: Design view gives you a more detailed view of the structure of the form. You can see the Header, Detail, and Footer sections for the form. The form is not actually running when it is shown in Design view. Therefore, you cannot see the underlying data while you are making design changes. However, there are certain tasks that you can perform more easily in Design view than in Layout view. You can:

- Add a wider variety of controls to the form, such as labels, images, lines, and rectangles.
- Edit text box control sources in the text boxes themselves, without using the property sheet.
- Resize form sections, such as the Form Header or the Detail section.
- Change certain form properties that cannot be changed in Layout view.

Modify a form in Layout view

After you create a form, you can easily fine-tune its design by working in Layout view. Using the actual form data as your guide, you can rearrange the controls and adjust their sizes. You can add new controls to the form and set the properties for the form and its controls.

- To switch to Layout view, right-click the form name in the Navigation Pane, and then click Layout View.
  - Access shows the form in Layout view.
- You can use the property sheet to modify the properties for the form and its controls and sections. To display the property sheet, press F4.
- You can use the **Field List** pane to add fields from the underlying table or query to the form design. To display the **Field List** pane, use one of the following methods:
  - On the **Format** tab, in the **Controls** group, click **Add Existing Fields**.
  - Press ALT+F8.
- You can drag fields directly from the **Field List** pane onto the form.
  - To add a single field, double-click it or drag it from the **Field List** pane to the section on the form where you want it displayed.
  - To add several fields at the same time, hold down CTRL and then click the fields that you want to add. Then, drag the selected fields onto the form.

### Modify a form in Design view
You can also fine-tune your form's design by working in Design view. You can add new controls and fields to the form by adding them to the design grid. The property sheet gives you access to many properties that you can set to customize the form.

- To switch to Design view, right-click the form name in the Navigation Pane, and then click **Design View**.
  - Access shows the form in Design view.
- You can use the property sheet to modify the properties for the form and its controls and sections. To display the property sheet, press F4.
- You can use the **Field List** pane to add fields from the underlying table or query to the form design. To display the **Field List** pane, use one of the following methods:
  - On the **Design** tab, in the **Tools** group, click **Add Existing Field**.
  - Press ALT+F8.
- You can drag fields directly from the **Field List** pane onto the form.
  - To add a single field, double-click it or drag it from the **Field List** pane to the section on the form where you want it displayed.
  - To add several fields at the same time, hold down CTRL and then click the fields that you want to add. Then, drag the selected fields onto the form.

### Understand layouts
Layouts are guides that align your controls horizontally and vertically to give your form a uniform appearance. You can think of a layout as a table, where each cell of the table is either empty or contains a single control. The following procedures show how to add, remove, or rearrange controls in layouts.

Layouts can be configured in many different ways, but are usually configured in a *tabular* or a *stacked* format. Within these two main formats, you can split or merge cells to customize the layout to better suit your controls.
• In tabular control layouts, controls are arranged in rows and columns like a spreadsheet, with labels across the top, as shown in the illustration.
  o Tabular control layouts always span two sections of a form. Whichever section the controls are in, the labels are in the section above.

• In stacked layouts, controls are arranged vertically like you might see on a paper form, with a label to the left of each control, as shown in the illustration.
  o Stacked layouts are always contained within a single section on the form.

Access automatically creates *stacked* layouts in either of the following circumstances:
- You create a new form by clicking **Form** in the **Forms** group on the **Create** tab.
- You create a new form by clicking **Blank Form** in the **Forms** group on the **Create** tab, and then dragging a field from the **Field List** pane to the form.

**Create a new layout**
On an existing form object, you can create a new control layout by doing the following:

1) Select a control that you want to add to the layout.
2) If you want to add other controls to the same layout, hold down the SHIFT key and also select those controls.
3) On the **Arrange** tab, in the **Table** group, click **Tabular** or **Stacked**.
   a) Access creates the layout and adds the selected controls to it.

**Switch a layout from tabular to stacked, or from stacked to tabular**
To switch an entire layout from one type of layout to the other:

1) Select a cell in the layout that you want to change.
2) On the **Arrange** tab, in the **Rows & Columns** group, click **Select Layout**.
3) On the **Arrange** tab, in the **Table** group, click the layout type you want (**Tabular** or **Stacked**).
   a) Access rearranges the controls into the layout type that you want.
Add rows or columns to a layout

1) Select a cell adjacent to where you want to add the new row or column.
2) On the Arrange tab, in the Rows & Columns group, click Insert Above or Insert Below to insert a new row above or below the current row.
   a) Click Insert Left or Insert Right to insert a new column to the left or right of the current column.
   b) Access adds the new row or column to the layout.

Remove rows or columns from a layout

1) Select a cell in the row or column that you want to remove.
2) On the Arrange tab, in the Rows & Columns group, click Select Column or Select Row to select the column or row.
3) Press the DELETE key.
   a) Access deletes the row or column from the layout.

Rearrange controls in a layout

You can move a control within a layout, or between layouts, by dragging it to the location that you want. As you drag the control, a horizontal or vertical bar indicates where it will be placed when you release the mouse button. If you drag the control over an empty cell, Access highlights the whole cell to indicate where the control will be placed.

You can also move a control within a layout by selecting it, holding down the ALT key, and then using the arrow keys to move the control.

If you use the ALT+arrow key combination to move a control to the left or right, and you move it outside the boundaries of the layout, Access adds new columns to the layout as needed to accommodate the move. To delete an empty column in a layout, in Layout view, select a cell in that column, and then press the DELETE key.

Add controls to a layout

Add a new field from the Field List pane to an existing control layout

1) Drag the field from the Field List pane to the layout.
   a) A horizontal or vertical bar indicates where the field will be placed when you release the mouse button.
   b) If you drag the field over an empty cell, Access highlights the whole cell to indicate where the field will be placed.

Add existing controls to an existing control layout

1) In Layout view, select the first control you want to add to the control layout.
   a) If you want to add other controls to the same layout, hold down the SHIFT key and also select those controls. You can select controls in other control layouts.
2) Drag the selected fields to the layout. A horizontal or vertical bar indicates where the fields will be placed when you release the mouse button.
Remove controls from a layout

Removing controls from a layout enables you to place the controls anywhere on the form without affecting the positioning of any other controls.

1) On the **Home** tab, in the **Views** group, click **Views**, and then click **Design view**.

2) Select the cells that you want to remove from the layout, including any labels.
   a) To select multiple cells, hold down the Shift key and then click the cells you want to remove.
   b) To select all of the cells in the layout, on the **Arrange** tab, in the **Rows & Columns** group, click **Select Layout**.

3) Right-click one of the selected cells, point to **Layout**, and then click **Remove Layout**.
   a) Access removes the selected cells from the layout.

4) If you left some controls in the layout, they will now overlap with the controls you removed. While the controls are still selected, drag the selected controls away from the layout, being careful not to drop the controls back into the layout.

Introduction to Controls

Controls are the parts of a form that you use to enter, edit, or display data. For example, a text box is a popular control for displaying data on reports, or for entering and displaying data on forms. Other popular controls include command buttons, check boxes, and combo boxes (drop-down lists).

Controls can be bound, unbound, or calculated:

- **Bound control**: A control whose source of data is a field in a table or query is called a bound control. You use bound controls to display values that come from fields in your database. The values can be text, dates, numbers, Yes/No values, pictures, or graphs. For example, a text box that displays an employee's last name might get this information from the Last Name field in the Employees table.

- **Unbound control**: A control that doesn't have a source of data (such as a field or expression) is called an unbound control. You use unbound controls to display information, pictures, lines or rectangles. For example, a label that displays the title of a form is an unbound control.

- **Calculated control**: A control whose source of data is an expression, rather than a field, is called a calculated control. You specify the value that you want to use as the source of data in the control by defining an *expression*. An expression can be a combination of operators (such as = and +), control names, field names, functions that return a single value, and constant values.
  o For example, the following expression calculates the price of an item with a 25 percent discount by multiplying the value in the Unit Price field by a constant value (0.75).
    
    \[\text{Price} = \text{Unit Price} \times 0.75\]

  o An expression can use data from a field in the form’s underlying table or query, or data from another control on the form.

Note: Calculations can also be performed in tables by adding a Calculated field, or in queries by entering an expression in the Field row of the query grid. You can then bind forms to those tables or queries, and the calculations appear on the forms without needing to create a calculated control.
When you create a form, it is probably most efficient to add and arrange all the bound controls first, especially if they make up most of the controls on the object. You can then add the unbound and calculated controls that complete the design by using the tools in the Controls group on the Design tab in Layout view or Design view.

You bind a control to a field by identifying the field from which the control gets its data. You can create a control that is bound to the selected field by dragging the field from the Field List pane to the form. The Field List pane displays the fields of the form's underlying table or query. To display the Field List pane, open the object in Layout view or Design view, and then on the Design tab, in the Tools group, click Add Existing Fields. When you double-click a field in the Field List pane, Access adds the appropriate type of control for that field to the object.

Alternatively, you can bind a field to a control by typing the field name in the control itself (if the object is open in Design view), or in the Control Source property box in the control's property sheet. The property sheet defines the characteristics of the control, such as its name, the source of its data, and its format. To display or hide the property sheet, press F4.

Using the Field List pane is the best way to create a bound control for two reasons:

- Access automatically fills in the control's attached label with the name of the field (or the caption defined for that field in the underlying table or query), so you don't have to type the control's label yourself.
- Access automatically sets many of the control's properties to the appropriate values according to the properties of the field in the underlying table or query (such as the Format, Decimal Places, and Input Mask properties).

If you already created an unbound control and want to bind it to a field, set the value in the control's Control Source property box to the name of the field.

Add a field to a form

You can quickly add fields to a form by using the Field List. When you double-click a field in the Field List (or if you drag a field from the list to your form), Microsoft Access 2010 automatically creates the appropriate control to display the field — for example, a text box or check box — and then binds the control to that field.

Add a field to a form by using the Field List

1) In the Navigation Pane, right-click the form that you want to change, and then click Layout View or Design View.
   a) If the Field List is not displayed, do one of the following:
      - On the Design tab, in the Tools group, click Add Existing Fields.
      - OR
      - Press ALT+F8.

2) In the Field List, find the table containing the field you want to add. To show or hide the list of fields in each table, click the plus (+) or minus sign (-) next to the table name.

   Note: If the database does not contain any tables, the Field List is empty.
3) When you find the field you want to add, do one of the following to add it to the form:
   a) Double-click the field.
   b) Drag the field from the Field List to the form.
   c) Hold the CTRL key and click several fields, and then drag them all to the form at the same time.
      
      **Note:** This technique can be used only on fields in the **Fields available for this view** section of the Field List.
   d) You can add a range of fields from the Field List to the form by doing the following:
      - Click the first field in the range.
      - Press and hold the SHIFT key, and then click the last field in the range.
        
        1. Access selects the fields you clicked, as well as all the fields in between.
      - Drag all the selected fields to the form.
      
      **Note:** This technique can only be used on fields in the **Fields available for this view** section of the Field List.

4) Access creates the appropriate control to display each field, and also binds the control to the field. In addition, Access creates an attached label for the control. You can edit the label by clicking it once to select it, and then clicking it again to place the insertion point in the label. Then, you can edit the label in the same way that you would edit text in a document.
Learn more about the Field List
The following list contains tips about working with the Field List that may be helpful to you as you add fields to your form. To display the Field List, open a form in Layout view or Design view, and then press ALT+F8.

- If the form’s Record Source property is empty, and you drag a field from the Field List to the form, Access automatically fills in the form’s Record Source property for you.
- If the form's Record Source property is a SELECT statement or a table name, you can toggle the Field List display between the following two states:
  - All database fields: Displays the Fields available for this view section, the Fields available in related tables section (if such tables exist), and the Fields available in other tables section (if such tables exist).
  - Record source fields only: Displays only the Fields available for this view section.

- To switch between the two modes, click either Show only fields in the current record source or Show all tables at the top of the Field List.

  **Note:** If you have set the form's Record Source property to a saved query, the Field List displays only the Fields available for this view section (there is no Show all tables option at the top of the Field List). To edit the record source, right-click any one of the fields, and then click Edit Record Source.
• If you have set the form’s **Record Source** property to a table, and you then drag a field from the **Fields available in related tables** section of the Field List to the form, Access changes the **Record Source** of the form to a SELECT statement that includes the field you are adding. If Access cannot determine how to relate the tables, it displays the **Choose Relationship** dialog box so you can choose the appropriate relationship.

• If you drag a field from the **Fields available in other tables** section of the Field List to the form, Access displays the **Specify Relationship** dialog box. You must use this dialog box to specify a relationship between the other table and the form’s record source. If you are not sure which fields to enter in this dialog box, you might want to click **Cancel** and then examine your table relationships.

It is possible that a field in the **Fields available in other tables** section of the Field List is indirectly related to the record source of the form through one of the tables in the **Fields available in related tables** section. If this is the case, dragging a field from one of the related tables to the form may cause one or more tables to move from the **Fields available in other tables** section to the **Fields available in related tables** section.

• If the record source of the form is a SELECT statement or a saved query, you can view and edit the record source in the Query Builder by right-clicking any field in the Field List and then clicking **Edit Record Source**. If the record source is a table, this command invokes the Query Builder on the table. Access asks you if you want to create a query based on the table; click **Yes** to change the record source to a query, or **No** to leave the record source unchanged.

• You can open a table in Datasheet view by clicking **Edit Table** next to the table name, or by right-clicking the table name in the Field List and then clicking **Open Table in Datasheet View**.

• You can undock the Field List task pane from the right side of the Access work area by positioning the pointer in the title bar of the Field List until it turns into a move pointer, and then dragging the Field List to the left. You can position the list anywhere in the work area, or you can dock it to the left side of the work area by dragging it all the way to the left.

### Add a field to a form without using the Field List

In general, using the Field List is the easiest way to add a field to a form. However, you can also add a field by first creating a control, and then binding it to a field. The following procedure shows you how.

1) In the Navigation Pane, right-click the form and then click **Layout View** or **Design View**.

2) On the **Design** tab, in the **Controls** group, click the down-arrow next to the Control Gallery.

   a) If **Use Control Wizards** is not highlighted, click it to highlight it.

   **Control Wizards**: You can use the Control Wizards to help you create command buttons, list boxes, subforms, combo boxes, and option groups.

   A Control Wizard asks you a series of questions about how you want the control to look and operate, and then it creates the control and sets the control’s properties based on your answers.

   b) If you would rather create controls without the help of the wizards, click **Use Control Wizards** so that it is not selected.

3) On the **Design** tab, in the **Controls** gallery, click the tool for the type of control you want to add.

   a) To determine the name of a tool, position the pointer over it. Access displays the name of the tool.
4) Click in the form where you want to position the control.
   a) If you have selected **Use Control Wizards** and the control you are placing has a wizard associated with it, the wizard starts and guides you through the settings for the control.
      - If you don't position the control perfectly on the first try, you can move it by doing the following:
        1. Click the control to select it. If there is a label associated with the control, hold down the CTRL key and then click the label to select it as well.
        2. Position the pointer over the control until it turns into a move pointer.
        3. Click and drag the control to the location that you want.
   b) If you use a control wizard, the wizard might contain steps to help you bind the control to a field. If the wizard does not bind the control to a field, and the control is of the type that can display data (a text box or combo box, for example), you must enter a field name or expression in the **Control Source** property for the control before it will display any data. Use the following procedure to bind a control to a field:
      - Display the property sheet for the control by clicking the control, and then pressing F4.
      - On the **Data** tab of the property sheet, click the drop-down arrow next to the **Control Source** property and select the field you want displayed in the control.
      - You can also type an expression in the **Control Source** box.

**Create a form by using the Form tool**

You can use the Form tool to quickly create a single item form. This type of form displays information about one record at a time, as shown in the illustration:

1) The form displays information for a single record.
2) In some cases, Access adds a subdatasheet to display related information.

When you use the Form tool, all the fields from the underlying data source are added to the form. You can start to use the new form immediately, or you can modify it in Layout view or Design view to better suit your needs.

**Create a single item form**

1) In the Navigation Pane, click the table or query that contains the data that you want to see on the form.
2) On the **Create** tab, in the **Forms** group, click **Form**.
   a) Access creates the form and displays it in Layout view. In Layout view, you can make design changes to the form while it is displaying data. For example, you can adjust the size of the text boxes.
3) To begin using the form, switch to Form view.
   a) On the **Home** tab, in the **Views** group, click **View**, and then click **Form View**.
If Access finds a single table that has a one-to-many relationship with the table or query that you used to create the form, Access adds a datasheet to the form that is based on the related table or query. For example, if you create a simple form that is based on the Employees table, and there is a one-to-many relationship that is defined between the Employees table and Orders table, the datasheet displays all the records in the Orders table that relate to the current Employee record.

If you decide that you do not want the subdatasheet on the form, you can delete the subdatasheet by switching to Layout view, selecting the datasheet, and then pressing DELETE.

If there is more than one table with a one-to-many relationship to the table that you used to create the form, Access does not add any datasheets to the form.

Create a form by using the Multiple Items tool
A multiple item form, also known as a continuous form, lets you show information from more than one record at a time. The data is arranged in rows and columns (similar to a datasheet), and multiple records are displayed at a time. However, because it is a form, there are more customization options than with a datasheet. You can add features such as graphical elements, buttons, and other controls.

Create a multiple item form
1) In the Navigation Pane, click the table or query that contains the data that you want to see on the form.
2) On the Create tab, in the Forms group, click Multiple Items. If Multiple Items is not available, click More Forms, and then click Multiple Items.
   a) Access creates the form and displays it in Layout view. In Layout view, you can make design changes to the form while it is displaying data. For example, you can adjust the size of the text boxes to fit the data.
3) To begin using the form, switch to Form view:
   a) On the Home tab, in the Views group, click View, and then click Form View.
      • Access creates the form and displays it in Layout view.

Create a form by using the Form Wizard
Access provides several quick-create form tools on the Create tab, each of which lets you create a form with a single click. However, if you want to be more selective about what fields appear on the form, you can use the Form Wizard instead. The wizard also lets you define how the data is grouped and sorted, and you can use fields from more than one table or query (provided that you specified the relationships between the tables and queries beforehand).

Start the Form Wizard
1) On the Create tab, in the Forms group, click More Forms, and then click Form Wizard.
2) Follow the directions on the pages of the Form Wizard.
   Note: If you want to include fields from multiple tables and queries on the form, do not click Next or Finish after you select the fields from the first table or query on the first page of the Form Wizard.
   Instead, repeat the steps to select a table or query, and then click any additional fields that you want to include on the form. Then click Next or Finish to continue.
3) On the last page of the wizard, click Finish.
The Form Wizard can create a variety of results depending on the options that you select. As a result, it’s recommended that you run the wizard several times, experimenting with different options each time, until you get the results that you want.

**Create a form by using the Blank Form tool**

If the wizard or the form-building tools don't meet your needs, you can use the Blank Form tool to build a form. This can be a very quick way to build a form, especially if you plan to put only a few fields on your form.

1) On the **Create** tab, in the **Forms** group, click **Blank Form**.
   a) Access opens a blank form in Layout view, and displays the **Field List** pane.
2) In the **Field List** pane, click the plus sign (+) next to the table or tables that contain the fields that you want to see on the form.
3) To add a field to the form, double-click it or drag it onto the form.

**Notes**
- After the first field has been added, you can add several fields at once by holding down the CTRL key, clicking several fields, and then dragging them onto the form at the same time.
- The order of the tables in the **Field List** pane can change, depending on which part of the form is currently selected. If the field you want to add is not visible, try selecting a different part of the form and then try adding the field again.
4) Use the tools in the **Header/Footer** group on the **Design** tab to add a logo, title, or the date and time to the form.
5) Use the tools in the **Controls** group of the **Design** tab to add a wider variety of controls to the form.
   a) For a slightly larger selection of controls, switch to Design view by right-clicking the form and then clicking **Design View**.

**Create a tabbed form**

Adding tabs to a form can make it more organized and easier to use, especially if the form contains many controls. By placing related controls on separate pages of the tab control, you can reduce clutter and make it easier to work with your data.

To add tabs to a form, you use the **Tab Control** tool. Each page of a tab control acts as a container for other controls, such as text boxes, combo boxes, or command buttons. The following procedures show you how to add a tab control to a form.

**Note:** All of the following procedures assume that a form has already been created, and that the form is open in Design view. To display a form in Design view, in the Navigation Pane, right-click the form, and then click **Design View**.
Add a tab control to a form
1) On the Design tab, in the Controls group, click the Tab Control tool.
2) Click on the form where you want to place the tab control.
   a) Access places the tab control on the form.

Move existing controls to a tab page
1) Select the controls that you want to move to the tab page. To select multiple controls, hold down the SHIFT key and then click the controls you want to move.
2) On the Home tab, in the Clipboard group, click Cut.
3) Click the label text on the tab that corresponds to the page on which you want to place the controls. A selection box will appear on the tab page.
   Note: If the selection box is not displayed on the tab page, the controls will not be correctly attached to the page when you perform the next step.
4) On the Home tab, in the Clipboard group, click Paste.
   a) Access places the control or controls on the tab page.
5) To confirm that the controls are correctly attached to the page, click another tab on the tab control.
   a) The controls you just pasted should disappear, and then reappear when you click the original tab.

Drag fields from the Field List task pane to a tab page
1) Select the tab page to which you want to add the fields.
2) On the Design tab, in the Tools group, click Add Existing Field.
3) Navigate to the table that contains the fields you want to add.
4) Drag each field from the Field List task pane to the tab page. Alternatively, select several fields by holding down the CTRL or SHIFT key while clicking the fields, and then drag them all to the tab page at the same time.
   Note: As you drag the fields over the tab page, the page turns black to indicate that the fields will be attached to that page.
5) Release the mouse button.
   a) Access places the field or fields on the tab page.

Add other controls to a tab page
You can add other types of controls, such as images, command buttons, and check boxes, to the pages of a tab control.
1) Select the tab page to which you want to add the control.
2) On the Design tab, in the Controls group, click the tool for the control that you want to add.
3) Move the pointer to the tab page.
   a) As you move the pointer over the page, it turns black to indicate that the control will be attached to that page.
4) Click on the tab page where you want to place the control.
   a) Access places the control on the tab page.
Add a new tab page
1) Select a tab, or click in the blank area at the top of the tab control to select the entire control.
2) On the Design tab, in the Controls group, click Insert Page. Alternatively, you can right-click the tab control and then click Insert Page.
   a) Access adds a new blank page at the end of the existing pages.

Reorder tab pages
1) Right-click a tab, or right-click the blank area at the top of the tab control.
2) Click Page Order.
3) In the Page Order dialog box, select the page that you want to move.
4) Click Move Up or Move Down to place the page in the order you want.
5) Repeat steps 3 and 4 for any other pages that you want to move.
   Note: The Page Order command is not to be confused with the Tab Order command. The Tab Order command is used to set the order in which the cursor advances through the controls on a form as you press the TAB key.

Rename a tab page
1) Click the tab that you want to rename.
2) If the Property Sheet task pane is not displayed, press F4 to display it.
3) On the All tab of the Property Sheet, modify the text in the Name property box, and then press ENTER.

Delete a tab page
1) Right-click the tab page that you want to delete, and then click Delete Page.
   a) Access deletes the page and all the controls that it contains.

Create a navigation form
Access 2010 includes a new Navigation Control that makes it easy to switch between the various forms in your database. A navigation form is simply a form that contains a Navigation Control. This section describes how to create and modify a navigation form, and how to set formatting and display options.

Create a navigation form
1) Open the database to which you want to add a navigation form.
2) On the Create tab, in the Forms group, click Navigation, and then select the style of navigation form that you want.
   a) Access creates the form, adds the Navigation Control to it, and displays the form in Layout view.

Add a form to a navigation form
1) If the Navigation Pane is not already displayed, press F11 to display it.
2) Ensure that the navigation form is open in Layout View by right-clicking it in the Navigation Pane, and then clicking Layout View.
3) Drag the form from the Navigation Pane to the [Add New] button.
   a) Access creates the new navigation button and displays the form in the object pane.
**Edit the label at the top of the navigation form**

When you create a new navigation form, Access adds the label “Navigation Form” to the form header by default. To edit this label:

1) Open the form in Layout View by right-clicking it in the Navigation Pane, and then clicking **Layout View**.
2) Click the label in the form header once to select it, and then again to place the cursor in the label.
3) Edit the label to suit your needs, and then press ENTER.

**Edit the form caption**

The form caption is the text that appears on the document tab above the form (or in the window title bar if you have set the database to display objects as overlapping windows). To edit the form caption:

1) Open the form in Layout View by right-clicking it in the Navigation Pane, and then clicking **Layout View**.
2) Right click on the form header near the top of the form, and then click **Form Properties**.
3) In the Property Sheet task pane, on the **All** tab, edit the **Caption** property to suit your needs.

**Set the navigation form as the default display form**

Because a navigation form is often used as a switchboard or “home page” for a database, it makes sense to display it by default each time that the database is opened.

Use this procedure to set the navigation form as the default display form.

1) On the **File** tab, under **Help**, click **Options**.
2) Click **Current Database**.
3) To set the default form that is displayed when you open the database in Access, under **Application Options**, select the form from the **Display Form** list.

**Create a split form**

A split form gives you two views of your data at the same time — a Form view and a Datasheet view. The two views are connected to the same data source and are synchronized with each other at all times. Selecting a field in one part of the form selects the same field in the other part of the form. You can add, edit, or delete data from either part (provided the record source is updateable and you have not configured the form to prevent these actions).

Working with split forms gives you the benefits of both types of forms in a single form. For example, you can use the datasheet portion of the form to quickly locate a record, and then use the form portion to view or edit
the record. The form portion can serve as an attractive and functional header for the datasheet portion. This technique is used in many of the template databases that are available for Access.

**Create a new split form by using the Split Form tool**

This procedure creates a new split form from scratch. The form is based on a table or query that you select from the Navigation Pane or have open in Datasheet view.

1) In the Navigation Pane, click the table or query that contains the data that you want on your form. Or, open the table or query in Datasheet view.

2) On the **Create** tab, in the **Forms** group, click **More Forms**, and then click **Split Form**.
   a) Access creates the form and displays it in Layout view.
   b) In Layout view, you can make design changes to the form while it is displaying data. For example, you can adjust the size of the text boxes to fit the data, if necessary.

**Turn your existing form into a split form**

You can turn your existing form into a split form by setting a few form properties.

1) Open your form in Layout view by right-clicking it in the Navigation Pane and then clicking **Layout View**.

2) If the property sheet is not already displayed, press F4 to display it.

3) Select **Form** from the drop-down list at the top of the property sheet.

4) On the **Format** tab of the property sheet, in the **Default View** drop-down list, select **Split Form**.

5) Save and close the form, and then double-click the form in the Navigation Pane to open it in Form view.

**Fine-tune your split form**

After you create your split form, you can make some adjustments in Design view to get the results you want.

To switch to Design view, right-click the form in the Navigation Pane and then click **Design View**.

**Set the split form properties**

The following table lists some of the properties that are related to split forms that you can set on the property sheet to fine-tune your form. If the property sheet is not already displayed, press F4 to display it. Also, ensure that **Form** is selected in the drop-down list at the top of the property sheet. All of the properties in the table are located on the **Format** tab of the property sheet. Default values can vary depending on how you create the split form.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Split Form Size</td>
<td>Allows you to specify an exact height or width (depending on whether the form is split vertically or horizontally) for the form part of the split form. For example, type 1&quot; to set the form to a height or width of 1 inch. Type Auto to set the dimension by other means, such as dragging the splitter bar in Layout view.</td>
</tr>
<tr>
<td>Split Form Orientation</td>
<td>Allows you to define whether the datasheet appears above, below, to the left, or to the right of the form.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Split Form</td>
<td>If set to Yes, Access allows you to resize the form and datasheet by moving the splitter bar that separates the two parts. You can drag the splitter bar to provide more room or less room for the datasheet. If this property is set to No, the splitter bar is hidden, and the form and datasheet cannot be resized.</td>
</tr>
<tr>
<td>Splitter Bar</td>
<td>If set to Yes, Access allows you to resize the form and datasheet by moving the splitter bar that separates the two parts. You can drag the splitter bar to provide more room or less room for the datasheet. If this property is set to No, the splitter bar is hidden, and the form and datasheet cannot be resized.</td>
</tr>
<tr>
<td>Split Form</td>
<td>If set to Allow Edits (and the form's record source is updateable), Access allows edits to be made on the datasheet. If the property is set to Read Only, Access prevents edits from being made on the datasheet.</td>
</tr>
<tr>
<td>Datasheet</td>
<td>If set to Allow Edits (and the form's record source is updateable), Access allows edits to be made on the datasheet. If the property is set to Read Only, Access prevents edits from being made on the datasheet.</td>
</tr>
<tr>
<td>Split Form</td>
<td>Allows you to define which portion of the form is printed when you print the form. If the property is set to Form Only, only the form portion is printed. If the property is set to Datasheet Only, only the datasheet portion is printed.</td>
</tr>
<tr>
<td>Printing</td>
<td>Allows you to define which portion of the form is printed when you print the form. If the property is set to Form Only, only the form portion is printed. If the property is set to Datasheet Only, only the datasheet portion is printed.</td>
</tr>
<tr>
<td>Save Splitter Bar</td>
<td>If set to Yes, the form opens with the splitter bar in the same position in which you last left it. If this property is set to No, the form and datasheet cannot be resized, and the splitter bar is hidden.</td>
</tr>
<tr>
<td>Position</td>
<td>If set to Yes, the form opens with the splitter bar in the same position in which you last left it. If this property is set to No, the form and datasheet cannot be resized, and the splitter bar is hidden.</td>
</tr>
</tbody>
</table>

**Fix the form separator in place**
To fix the form separator in a certain position so that it cannot be moved by the user, do the following:

1) Switch to Design view by right-clicking the form in the Navigation Pane, and then clicking Design View.
2) If the property sheet is not already displayed, press F4 to display it.
3) Select Form from the drop-down list at the top of the property sheet.
4) On the Format tab of the property sheet, set the Split Form Splitter Bar property to No.
5) Set the Save Splitter Bar Position property to Yes.
6) Switch to Layout view by right-clicking the form in the Navigation Pane, and then clicking Layout View.
7) Drag the splitter bar to the position that you want (or type an exact height in the Split Form Size property box).
8) Switch to Form view to view the results.
    a) The separator is fixed in the position where you set it, and the splitter bar is hidden.

**Add a field to a split form**
1) Switch to Layout view by right-clicking the form in the Navigation Pane, and then clicking Layout View.
2) If the Field List pane is not displayed, on the Design tab, in the Tools group, click Add Existing Fields.
3) Locate the field that you want to add, and then drag the field to the form.
   Note: If you drag the field to the datasheet, the field is also added to the form, but it might not appear where you want it. Therefore, we recommend dragging the field to the form.

**Delete a field from a split form**
1) Switch to Layout view by right-clicking the form in the Navigation Pane, and then clicking Layout View.
2) In the form portion of the split form, click the field to select it, and then press DELETE.
   a) The field is removed from both the form and the datasheet.
Create a form that contains a subform
When you are working with relational data (related data that is stored in separate tables), you often need to view data from multiple tables or queries on the same form. For example, you might want to see a customer record from one table and information about that customer’s orders from another table at the same time. Subforms are a convenient tool for doing this, and Microsoft Access 2010 provides several ways to help you create them quickly.

Overview of subforms
A subform is a form that is inserted in another form. The primary form is called the main form. A form/subform combination is sometimes referred to as a hierarchical form, a master/detail form, or a parent/child form.

Subforms are especially effective when you want to show data from tables or queries that have a one-to-many relationship. A one-to-many relationship is an association between two tables in which the primary key value of each record in the primary table corresponds to the value in the matching field or fields of many records in the related table. For example, you can create a form that displays employee data, and contains a subform that displays each employee's orders. The data in the Employees table is the "one" side of the relationship. The data in the Orders table is the "many" side of the relationship (each employee can have more than one order).

The main form and subform are linked so that the subform displays only records that are related to the current record in the main form. For example, when the main form displays Nancy Freehafer’s information, the subform displays only her orders. If the form and subform were unlinked, the subform would display all the orders, not just Nancy's.

1 The main form shows data from the "one" side of the relationship.
2 The subform shows data from the "many" side of the relationship.
The following table defines some of the terminology that is associated with subforms. Access will handle most of the details if you use the procedures in this section, but it is helpful to know what is occurring behind the scenes if you need to make modifications later.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subform control</td>
<td>The control that embeds a form into a form. You can think of the subform control as a &quot;view&quot; of another object in your database, whether it is another form, a table, or a query. The subform control provides properties which allow you to link the data displayed in the control to the data on the main form.</td>
</tr>
<tr>
<td>Source Object property</td>
<td>The property of the subform control that determines what object is displayed in the control.</td>
</tr>
<tr>
<td>Datasheet</td>
<td>A simple display of data in rows and columns, much like a spreadsheet. The subform control displays a datasheet when its source object is a table or query, or when its source object is a form whose Default View property is set to Datasheet. In these cases, the subform is sometimes referred to as a datasheet or subdatasheet instead of as a subform.</td>
</tr>
<tr>
<td>Link Child Fields property</td>
<td>The property of the subform control that specifies which field or fields in the subform link the subform to the main form.</td>
</tr>
<tr>
<td>Link Master Fields property</td>
<td>The property of the subform control that specifies which field or fields on the main form link the main form to the subform.</td>
</tr>
</tbody>
</table>

Notes

- For best results, establish table relationships before following the procedures in this section. This enables Access to automatically create the links between subforms and main forms. To view, add, or modify relationships between the tables in your database, on the Database Tools tab, in the Show/Hide group, click Relationships.
- When a subform control has a form as its source object, it contains the fields that you place on the form, and it can be viewed as a single form, continuous form, or datasheet. One advantage of basing a subform on a form object is that you can add calculated fields to the subform, such as \([\text{Quantity}] \times [\text{Unit price}]\).
- You can insert a datasheet or pivot view into a form by creating a subform control whose source object is a table or query.
Create or add a subform

Use the following table to determine which procedure is most appropriate for your situation.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Recommended procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want Access to create both a main form and a subform, and to link</td>
<td>Create a form that contains a subform by using the Form Wizard</td>
</tr>
<tr>
<td>the subform to the main form.</td>
<td></td>
</tr>
<tr>
<td>You want to use an existing form as the main form, but you want Access</td>
<td>Add one or more subforms to an existing form by using the Subform Wizard</td>
</tr>
<tr>
<td>to create a new subform and add it to the main form.</td>
<td></td>
</tr>
<tr>
<td>You want to use an existing form as the main form, and you want to</td>
<td>Create a subform by dragging one form onto another</td>
</tr>
<tr>
<td>add one or more existing forms to that form as subforms.</td>
<td></td>
</tr>
</tbody>
</table>

Create a form that contains a subform by using the Form Wizard

This procedure creates a new form and subform combination by using the Form Wizard. This is also the quickest way to get started if you have not already created the forms that you want to use as the main form or the subform.

1) On the Create tab, in the Forms group, click Form Wizard.
2) On the first page of the wizard, in the Tables/Queries drop-down list, select a table or query. For this example, to create an Employees form that displays orders for each employee in a subform, we will select Table: Employees (the "one" side of the one-to-many relationship).
   Note: It does not matter which table or query you choose first.
3) Double-click the fields that you want to include from this table or query.
4) On the same page of the wizard, in the Tables/Queries drop-down list, select another table or query from the list. For this example, we will select the Orders table (the "many" side of the one-to-many relationship).
5) Double-click the fields that you want to include from this table or query.
6) When you click Next, assuming that you set up the relationships correctly before you started the wizard, the wizard asks How do you want to view your data? - that is, by which table or query. Select the table on the "one" side of the one-to-many relationship. For this example, to create the Employees form, we will click by Employees. The wizard displays a small diagram of a form. The page should resemble the illustration.
   Note: If the wizard does not ask How do you want to view your data?, that means that Access did not detect a one-to-many relationship between the tables or queries that you selected. The wizard will continue, but Access will not add a subform to the form. You might want to click Cancel and examine your table relationships.

The box in the lower portion of the form diagram represents the subform.
7) At the bottom of the wizard page, select Form with subform(s), and then click Next.

8) On the What layout would you like for your subform? page, click the layout option that you want, and then click Next. Both layout styles arrange the subform data in rows and columns, but a tabular layout is more customizable. You can add color, graphics, and other formatting elements to a tabular subform, whereas a datasheet is more compact, like the datasheet view of a table.

9) On the last page of the wizard, type the titles that you want for the forms. Access names the forms based on the titles that you type, and labels the subform based on the title that you type for the subform.

10) Specify whether you want to open the form in Form view, so that you can view or enter information, or in Design view, so that you can modify its design, and then click Finish.

   a) Access creates two forms - one for the main form that contains the subform control, and one for the subform itself.

Add one or more subforms to an existing form by using the SubForm Wizard

Use this procedure to add one or more subforms to an existing form. For each subform, you can choose to have Access create a new form or use an existing form as the subform.

1) Right-click the existing form in the Navigation Pane, and then click Design View.

2) On the Design tab, in the Controls group, click the down-arrow to display the Controls gallery, and ensure that Use Control Wizards is selected.

3) On the Design tab, in the Controls gallery, click the Subform/Subreport button.

4) Click on the form where you want to place the subform.

5) Follow the directions in the wizard.

6) When you click Finish, Access adds a subform control to your form.

   a) If you chose to have Access create a new form for the subform instead of using an existing form, Access creates the new form object and adds it to the Navigation Pane.

Create a subform by dragging one form onto another

Use this procedure if you want to use an existing form as a main form, and you want to add one or more existing forms to that form as subforms.

1) In the Navigation Pane, right-click the form that you want to use as the main form, and then click Layout View.

2) Drag the form that you want to use as the subform from the Navigation Pane onto the main form.

   a) Access adds a subform control to the main form and binds the control to the form that you dragged from the Navigation Pane.

   b) Access also tries to link the subform to the main form, based on the relationships that have been defined in your database.

3) Repeat this step to add any additional subforms to the main form.

4) To verify that the linking was successful, on the Home tab, in the Views group, click View, click Form View, and then use the main form's record selector to advance through several records. If the subform filters itself correctly for each employee, then the procedure is complete.
If the previous test does not work, Access was unable to determine how to link the subform to the main form, and the Link Child Fields and Link Master Fields properties of the subform control are blank. You must set these properties manually by doing the following:

1) Right-click the main form in the Navigation Pane, and then click Layout View.
2) Click the subform control one time to select it.
3) If the Property Sheet task pane is not displayed, press F4 to display it.
4) In the Property Sheet, click the Data tab.
5) Click the Build button next to the Link Child Fields property box.
   a) The Subform Field Linker dialog box appears.
6) In the Master Fields and Child Fields drop-down lists, select the fields that you want to link the forms with, and then click OK.
   a) If you are not sure which fields to use, click Suggest to have Access try to determine the linking fields.
   Tip: If you do not see the field that you want to use to link the forms, you might need to edit the record source of the master form or child form to help make sure that the linking field is in it. For example, if the form is based on a query, you should make sure that the linking field is present in the query results.
7) Save the main form, switch to Form view, and then verify that the form works as expected.

**Change the default view of a subform**

When you add a subform to a form, the subform/subreport control displays the subform according to the subform’s Default View property. This property can be set to the following values:

- Single Form
- Continuous Forms
- Datasheet
- PivotTable
- PivotChart
- Split Form

When you first create a subform, this property may be set to Continuous Forms or perhaps Single Form. However, if you set the Default View property of a subform to Datasheet, then the subform will display as a datasheet on the main form.

To set the Default View property of a subform:

1) Close any open objects.
2) In the Navigation Pane, right-click the subform and then click Layout View.
3) If the Property Sheet is not already displayed, press F4 to display it.
4) In the drop-down list at the top of the Property Sheet, make sure Form is selected.
5) On the Format tab of the Property Sheet, set the Default View property to the view you want to use.
   Note: If the view you want to use is not in the list, make sure the “Allow…View” property for the view is set to Yes. For example, if you want to specify Datasheet as the default view for the form, make sure that the Allow Datasheet View property is set to Yes.
6) Save and close the subform, and then open the main form to check the results.
Add related data to a form without creating a subform

It is not always necessary to create a separate form object to display related data. For example, if you are working on a form in Layout view or Design view and you drag a table or query from the Navigation Pane to the form, Access creates a subform/subreport control that displays the data in that object. The object’s Default View property determines how the data is displayed. Usually, this is set to Datasheet view, but you can also set the Default View property of a table or query to PivotTable or PivotChart, giving you more flexibility in displaying related data on forms.

Note: Changing the Default View property for a table or query determines how it is displayed whenever it is opened, whether you open it from the Navigation Pane or view it in a subform/subreport control. Because changing the view settings for a table can sometimes cause confusion when opening the table from the Navigation Pane, we recommend using a query for this procedure instead of a table.

Set the tab order for controls

When you use a form, you can switch between controls by pressing the TAB key. You can specify the order in which the controls on a form respond to the TAB key. On a well-designed form, the controls respond to the TAB key in a logical order — for example, from top to bottom and from left to right — so that the form is easier to use.

Change the tab order for controls

When you use tools such as wizards or layouts to create a form, Access sets the tab order to the same order that the controls appear on the form: top to bottom and left to right. However, if you create or modify a form without using a wizard or a layout, you might create a tab order that does not correspond to the location of the controls on the form. Use the following procedure to change the tab order of the controls.

1) In the Navigation Pane, right-click the form and then click Design View.
2) On the Design tab, in the Tools group, click Tab Order.
3) In the Tab Order dialog box, under Section, click the section you want to change.
4) Do one of the following:
   a) If you want Access to create a top-to-bottom and left-to-right tab order, click Auto Order.
   b) If you want to create your own custom tab order, click the selector for the control you want to move.
      • Click and drag to select more than one control at a time.
      • Click the selector again and drag the control to the desired location in the list.
5) Click OK.

Remove a control from the tab order

1) In the Navigation Pane, right-click the form and then click Design View.
2) If the Property Sheet task pane is not displayed, press F4 to display it.
3) Select the control that you want to remove from the tab order.
4) On the Other tab of the Property Sheet, in the Tab Stop property box, click No.

Note: If you remove a control from the tab order, you can still click the control to select it as long as its Enabled property is set to Yes.
Change the tab behavior for the last field in the tab order

By default, when you press the TAB key while the cursor is in the last field in the tab order, Access displays the next record in the data source, and moves the cursor to the first field of the tab order. You can change this behavior so that Access stays on the current record. If the form contains page breaks, you can also change the behavior so that Access does not move the cursor to the next page. This is helpful if you want to restrict the user from editing more than one record (or page) at a time.

1) In the Navigation Pane, right-click the form and then click Design View.
2) If the Property Sheet task pane is not displayed, press F4 to display it.
3) Select Form from the drop-down list at the top of the Property Sheet.
4) On the Other tab of the Property Sheet, in the Cycle property box, select one of the following options.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Records</td>
<td>When you press TAB in the last field in the record, the focus moves to the first field in the next record.</td>
</tr>
<tr>
<td>Current Record</td>
<td>When you press TAB in the last field in the record, the focus moves back to the first field in the current record.</td>
</tr>
<tr>
<td>Current Page</td>
<td>When you press TAB in the last field in the form page, the focus moves back to the first field in the current page.</td>
</tr>
</tbody>
</table>

Note: You can divide a form into pages by using the Page Break control. This control is available in Design View, in the Controls group of the Design tab. If the form does not contain any page breaks, the Current Page option behaves in the same way as the Current Record option.

Set the default form that appears when you open a database

In many databases, it is helpful to have the same form open automatically every time that you open the database. To set the default startup form, you specify the form in Access Options. This procedure assumes that you have a database open and that you have already created the form that you want to display.

1) Click the File tab, and then under Help, click Options.
2) Click Current Database.
3) Under Application Options, in the Display Form list, select the form that you want to display when the database starts.

Note: To bypass this option and other startup options when opening the database in Access, hold down the SHIFT key while you open the database.

Create a user interface (UI) macro

In Microsoft Access 2010, macros (An action or set of actions that you can use to automate tasks.) that are attached to user interface (UI) objects such as command buttons, text boxes, forms, and reports are known as UI macros. This distinguishes them from data macros, which are attached to tables. You use (UI) macros to automate a series of actions (The basic building block of a macro; a self-contained instruction that can be combined with other actions to automate tasks. This is sometimes called a command in other macro languages.), such as opening another object, applying a filter, starting an export operation, and many other
tasks. This section introduces you to the newly-redesigned macro builder, and shows you the basic tasks involved in creating a UI macro.

**Overview**

Macros can be contained in macro objects (sometimes called standalone macros), or they can be embedded into the event properties of forms, reports, or controls. Embedded macros become part of the object or control in which they are embedded. Macro objects are visible in the Navigation Pane, under **Macros**; embedded macros are not.

Each macro is made up of one or more macro actions. Depending on the context you are working in, some macro actions may not be available for use.

**What’s New**

Access 2010 features a new designer for creating macros. Some of the benefits of this new design include:

- **Action Catalog**: Macro actions are organized by type and searchable.
- **IntelliSense**: When typing expressions, IntelliSense suggests possible values and lets you select the correct one.
- **Keyboard shortcuts**: Use key combinations for faster, easier macro writing.
- **Program flow**: Create more readable macros with comment lines and groups of actions.
- **Conditional statements**: Allow for more complex logic execution with support for nested If/Else/Else If.
- **Macro reuse**: The Action Catalog displays other macros you have created, letting you copy them into the one you’re working on.

**Create a standalone macro**

This procedure creates a standalone macro object that will appear under **Macros** in the Navigation Pane. Standalone macros are useful when you want to reuse the macro in many places of the application. By calling the macro from other macros, you can avoid duplicating the same code in multiple places.

1) On the **Create** tab, in the **Macros & Code** group, click **Macro**.
   a) Access opens the Macro Builder.
2) On the Quick Access Toolbar, click **Save**.
3) In the **Save As** dialog box, type a name for the macro, and then click **OK**.
4) Continue with the section “Add actions to a macro”.

---

Handout: Access 2010: Creating Forms
ICT Training, Maxwell School of Syracuse University

Topics came directly from Access 2010 Help.
Page 29
Create an embedded macro

This procedure creates a macro that is embedded in an event property of an object. Such a macro does not appear in the Navigation Pane, but can be called from events such as On Load or On Click.

Because the macro becomes part of the form or report object, embedded macros are recommended for automating tasks that are specific to a particular form or report.

1) In the Navigation Pane, right-click the form or report that will contain the macro, and then click Layout View or Design View.
   a) If the property sheet is not already displayed, press F4 to display it.

2) Click the control or section that contains the event property in which you want to embed the macro.
   a) You can also select the control or section (or the entire form) by using the drop-down list under Selection Type at the top of the property sheet.

3) In the Property Sheet task pane, click the Event tab.

4) Click in the property box for the event you want to trigger the macro.
   a) For example, for a command button, if you want the macro to run when the button is clicked, click in the On Click property box.

Notes

- If the property box contains the words [Embedded Macro], this means a macro has already been created for this event. You can edit the macro by continuing with the remaining steps in this procedure.

- If the property box contains the words [Event Procedure], this means that a Visual Basic for Applications (VBA) procedure has already been created for this event.
   o Before you can embed a macro in the event, you will need to remove the procedure. You can do this by deleting the words [Event Procedure], but you should first examine the event procedure to make sure that removing it will not break needed functionality in the database. In some cases, you can recreate the functionality of the VBA procedure by using an embedded macro.

5) Click the Build button.

6) If the Choose Builder dialog box appears, make sure Macro Builder is selected, and then click OK.
   a) Access opens the Macro Builder. Continue with the next section to add actions to the macro.

Add actions to a macro

Actions are the individual commands that make up a macro, and each is named according to what it does. For example, FindRecord or CloseDatabase.

Step 1: Browse or search for a macro action

The first step in adding an action is finding it in the Add New Action drop-down list or in the Action Catalog.

Notes

- By default, the Add New Action drop-down list and the Action Catalog only display the actions that will execute in non-trusted databases. To see all actions:
   o On the Design tab, in the Show/Hide group, click Show All Actions.
- If the Action Catalog is not displayed, on the Design tab, in the Show/Hide group, click Action Catalog.

To find an action, use one of the following methods:
• Click the arrow in the **Add New Action** drop-down list, and scroll down to find the action. Program flow elements are listed first, and then the macro actions are listed alphabetically.

• Browse for the action in the Action Catalog pane. The actions are grouped by category. Expand each category to view the actions. If you select an action, a short description of the action appears at the bottom of the Action Catalog.

• Search for the action in the Action Catalog pane by typing in the Search box at the top of the pane. As you type, the list of actions is filtered to show all macros that contain that text. Access searches both the macro names and their descriptions for the text you enter.

**Step 2: Add an action to a macro**

Once you have found the macro action you want, add it to the macro by using one of these methods:

• Select an action in the **Add New Action** list, or just begin typing the action name in the box.
  o Access adds the action at the point where the **Add New Action** list was displayed.

• Drag the action from the Action Catalog to the macro pane.
  o An insertion bar appears to show you where the action will be inserted when you release the mouse button.

• Double-click the action in the Action Catalog.
  o If an action is selected in the macro pane, Access adds the new action just below the selected one.
  o If a **Group, If, Else If, Else, or Submacro** block is selected in the macro pane, Access adds the new action to that block.
  o If no action or block is selected in the macro pane, Access adds the new action to the end of the macro.

**Notes**

• If you have already created one or more macros, they are listed under the **In this Database** node in the Action Catalog.
  o Dragging a standalone macro (one that is listed under **Macros**) into the macro pane creates a **RunMacro** action that runs the macro you dragged in. You can then use the drop-down list to call submacros, if present.
  o If you just want to copy the actions from a standalone macro into the current macro (instead of creating a **RunMacro** action), right-click it in the Action Catalog, and then click **Add Copy of Macro**.
  o Dragging an embedded macro (one that is listed under a form or report object) into the macro pane copies the actions from that macro into the current macro.

• You can also create an action by dragging a database object from the Navigation Pane to the macro pane. If you drag a table, query, form, report, or module to the macro pane, Access adds an action that opens the table, query, form, or report. If you drag another macro into the macro pane, Access adds an action that runs the macro.
**Step 3: Fill in arguments**

Most macro actions require at least one argument. You can view a description of each argument by selecting the action and then moving the pointer over the arguments. For many arguments, you can select a value from a drop-down list. If the argument requires you to type in an expression, IntelliSense helps you enter the expression by suggesting possible values as you type, as shown in the following illustration:

When you see a value that you want to use, add it to your expression by double-clicking it or using the arrow keys to highlight it and then pressing the TAB or ENTER key.

**Move an action**

Actions are executed in order, from the top to the bottom of the macro. To move an action up or down in the macro, use one of the following methods:

- Drag the action up or down to where you want it.
- Select the action, and then press CTRL + UP ARROW or CTRL + DOWN ARROW.
- Select the action, and then click the **Move Up** or **Move Down** arrow on the right side of the macro pane.

**Delete an action**

To delete a macro action:

- Select the action, and then press the DELETE key. Alternatively, you can click the **Delete** (X) button on the right side of the macro pane.

**Notes**

- If you delete a block of actions, such as an **If** block or a **Group** block, all the actions in the block are deleted as well.
- The **Move up**, **Move down**, and **Delete** commands are also available on the shortcut menu that appears when you right-click a macro action.

**Run a macro**

You can run a macro by using any of the following methods:

- Double-click the macro in the Navigation Pane.
- Call the macro by using the **RunMacro** or **OnError** macro action.
- Enter the macro name in an Event property of an object. The macro will run when that event is triggered.
Use expressions in form and report controls
When you use an expression as the data source for a control, you create a calculated control. For example, suppose that you have a report that displays multiple inventory records, and you want to create a total in the report footer that sums all the line items on the report.

To calculate the total, place a text box control in the report footer, and then set the ControlSource property of the text box to the following expression:

\[ \text{=Sum([table_field])} \]

In this case, \textit{table_field} is the name of the field that contains your subtotal values. That field can reside in a table or a query. The \textbf{Sum} function calculates the total for a set of values from your record source.

The following procedure shows how to enter an expression in a text box control.

Enter an expression in a text box control

1) In the Navigation Pane, right-click the form that you want to change, and then click Layout View or Design View on the shortcut menu.
2) Select the control into which you want to enter an expression.
   a) If the Property Sheet is not already displayed, press F4 to display it.
3) To manually create an expression, on the Data tab in the property sheet, click the ControlSource property of the text box, and then type \( \text{=} \) followed by the rest of your expression.
   a) For example, \( \text{=}\text{Sum([table_field])} \), making sure that you substitute the name of your field for \textit{table_field}.
4) To create an expression by using the Expression Builder, click the Build button \( \text{=} \) in the property box.

Guide to expression syntax
When you use Microsoft Access 2010, you often encounter situations in which you need to work with values that do not reside directly in your data. For example, you might need to calculate sales tax on an order, or calculate the total value of the order itself. You can calculate these values by using expressions.

To use expressions, you must first write them by using proper syntax. Syntax is the set of rules by which the words and symbols in an expression are correctly combined.

Think of it this way: when you want Access to do something, you have to speak its language. For example, suppose you want to tell Access "Look at the BirthDate field in the Customers table and tell me the year of the customer's birth." You can write this expression as \textbf{DatePart("yyy",[Customers]![BirthDate])}. This expression consists of the DatePart function and two argument values: "yyy," and [Customers]![BirthDate].
Let's examine that expression in more detail.

1. **DatePart** is a function that examines a date and returns a specific portion.
2. The *interval* argument tells Access which part of the date to return — in this case, "yyyy" tells Access that you want only the year part of the date returned.
3. The *date* argument tells Access where to look for the date value — in this case, [Customers]![BirthDate] tells Access to look for the date in the BirthDate field of the Customers table.

As you can see, expressions in Access use a language that is, initially, a little bit hard to read. With a good understanding of expression syntax and a little practice, it becomes much easier.

There are a few key concepts that you should understand before you try to write expressions. This section introduces the concepts that you need to understand to use proper syntax, and discusses the syntax that you use in expressions.

This section does not address Structured Query Language (SQL) syntax, nor is it intended as a guide to Visual Basic for Applications (VBA) syntax.

**Introduction**

To build an expression, you combine identifiers by using functions, operators, and constants. Any valid expression must contain at least one function or at least one identifier, and can also contain constants or operators. You can also use an expression as part of another expression — typically as an argument of a function.

- **Identifiers in expressions**: The general form of an identifier in an expression is `[Collection name]![Object name].[Property name].`
  - **Note**: You only have to specify enough parts of an identifier to make it unique in the context of your expression. It is not uncommon for an identifier to take the form `[Object name]`.

- **Functions in expressions**: The general form of an expression that uses a function is `Function(argument, argument)`, where one of the arguments is usually an identifier or an expression.
  - **Note**: Some functions do not require arguments.

Before you use a particular function, review the corresponding Help article for specific information about that function's syntax.

- **Operators in expressions**: The general form of an expression that uses an operator is `Identifier operator identifier`. There are exceptions to this form, as detailed in the tables shown in the Operators section.

- **Constants in expressions**: The general form of an expression that uses a constant is `Identifier comparison_operator constant`. 

---

Let's examine that expression in more detail.

1. **DatePart** is a function that examines a date and returns a specific portion.
2. The *interval* argument tells Access which part of the date to return — in this case, "yyyy" tells Access that you want only the year part of the date returned.
3. The *date* argument tells Access where to look for the date value — in this case, [Customers]![BirthDate] tells Access to look for the date in the BirthDate field of the Customers table.

As you can see, expressions in Access use a language that is, initially, a little bit hard to read. With a good understanding of expression syntax and a little practice, it becomes much easier.

There are a few key concepts that you should understand before you try to write expressions. This section introduces the concepts that you need to understand to use proper syntax, and discusses the syntax that you use in expressions.

This section does not address Structured Query Language (SQL) syntax, nor is it intended as a guide to Visual Basic for Applications (VBA) syntax.

**Introduction**

To build an expression, you combine identifiers by using functions, operators, and constants. Any valid expression must contain at least one function or at least one identifier, and can also contain constants or operators. You can also use an expression as part of another expression — typically as an argument of a function.

- **Identifiers in expressions**: The general form of an identifier in an expression is `[Collection name]![Object name].[Property name].`
  - **Note**: You only have to specify enough parts of an identifier to make it unique in the context of your expression. It is not uncommon for an identifier to take the form `[Object name]`.

- **Functions in expressions**: The general form of an expression that uses a function is `Function(argument, argument)`, where one of the arguments is usually an identifier or an expression.
  - **Note**: Some functions do not require arguments.

Before you use a particular function, review the corresponding Help article for specific information about that function's syntax.

- **Operators in expressions**: The general form of an expression that uses an operator is `Identifier operator identifier`. There are exceptions to this form, as detailed in the tables shown in the Operators section.

- **Constants in expressions**: The general form of an expression that uses a constant is `Identifier comparison_operator constant`. 

---
Objects, collections, and properties

All of the tables, queries, forms, reports, and fields in an Access database are individually known as objects. Every object has a name. Some objects are already named, such as the Contacts table in a database created from the Contacts database template. When you create a new object, you give it a name.

The set of all members of a particular type of object is known as a collection. For example, the set of all tables in a database is a collection. Some objects that are a member of a collection in your database can also be collections that contain other objects. For example, a table object is a collection that contains field objects.

Objects have properties, which describe, and provide a way to change, the object’s characteristics. For example, a query object has a Default View property that both describes and lets you specify how the query will appear when you run it.

Identifiers

When you use an object, collection, or property in an expression, you refer to that element by using an identifier. An identifier includes the name of the element that you are identifying and also the name of the element to which it belongs. For instance, the identifier for a field includes the name of the field and the name of the table to which the field belongs. An example of such an identifier is found in the preceding example of an expression: [Customers]![BirthDate].

In some cases, the name of an element will work by itself as an identifier. This is true when the name of the element is unique in the context of the expression that you are creating. The rest of the identifier is implied by the context. For example, if you are designing a query that uses only one table, the field names alone will work as identifiers, because the field names in a table must be unique in that table. Because you are using only one table, the table name is implied in any identifier that you use in the query to refer to a field.

In other cases, you must be explicit about the parts of an identifier for a reference to work. This is true when an identifier is not unique in the context of the expression. When there is ambiguity, you must explicitly denote enough parts of the identifier to make it unique in context. For example, suppose you are designing a query that uses a table named Products and a table named Orders, and both tables have a field named ProductID. In such a case, the identifier that you use in the query to refer to either ProductID field must include the table name in addition to the field name — for example, [Products]![ProductID].

Identifier operators: There are three operators that you can use in an identifier.

- The bang operator (!)
- The dot operator (.)
- The square brackets operator ([ ])

You use these operators by surrounding each part of the identifier with square brackets, and then joining them by using either a bang or a dot operator. For example, an identifier for a field named Last Name in a table named Employees can be expressed as [Employees]![Last Name]. The bang operator tells Access that what follows is an object that belongs to the collection that precedes the bang operator. In this case, [Last Name] is a field object that belongs to the collection [Employees], which is itself a table object.

Note: Strictly speaking, you do not always have to type square brackets around an identifier or partial identifier. If there are no spaces or other special characters in the identifier, Access automatically adds the brackets when it reads the expression. However, it is a good practice to type the brackets yourself — this helps you to avoid errors, and also functions as a visual clue that a particular part of an expression is an identifier.
Functions, operators, and constants

To create an expression, you need more than identifiers — you need to perform an action of some sort. You use functions, operators, and constants to perform actions in an expression.

Functions

A function is a procedure that you can use in an expression. Some functions, such as Date, do not require any input in order for them to work. Most functions, however, do require input, called arguments. In the example at the beginning of this section, the DatePart function uses two arguments: an interval argument (with a value of "yyyy") and a date argument (with a value of [Customers]!BirthDate). The DatePart function requires at least these two arguments (interval and date), but can accept up to four arguments.

Note: When a function has more than one argument, those arguments are separated by commas.

The following list shows some functions that are commonly used in expressions. Click the link for each function for more information about the syntax that you use with that function.

- **Date**: The Date function is used to insert the current system date into an expression. It is commonly used in conjunction with the Format function, and is also used in conjunction with field identifiers for fields that contain date/time data.

- **DatePart**: The DatePart function is used to determine or extract part of a date — usually a date that is obtained from a field identifier, but sometimes a date value that is returned by another function, such as Date.

- **DateDiff**: The DateDiff function is used to determine the difference between two dates — usually between a date that is obtained from a field identifier and a date that is obtained by using the Date function.

- **Format**: The Format function is used to apply a format to an identifier, and also to apply a format to the results of another function.

- **IIf**: The IIf function is used to evaluate an expression as true or false, and then return one specified value if the expression evaluates as true, and a different specified value if the expression evaluates as false.

- **InStr**: The InStr function is used to search for the position of a character or string within another string. The string that is searched is usually obtained from a field identifier.

- **Left, Mid, and Right**: These functions are used to extract characters from a string, starting with the leftmost character (Left), a specific position in the middle (Mid), or with the rightmost character (Right). They are commonly used in conjunction with the InStr function. The string from which these functions extract characters is usually obtained from a field identifier.

Operators

An operator is a word or symbol that indicates a specific arithmetic or logical relationship between the other elements of an expression. Operators can be:

- Arithmetic, such as the plus sign (+).
- Comparison operators, such as the equal sign (=).
- Logical operators, such as Not.

Operators are generally used to indicate a relationship between two identifiers. The following tables describe the operators that you can use in Access expressions.
**Arithmetic operators**

You use the arithmetic operators to calculate a value from two or more numbers or to change the sign of a number from positive to negative.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Sum two numbers.</td>
<td>[Subtotal]+[SalesTax]</td>
</tr>
<tr>
<td>-</td>
<td>Find the difference between two numbers or indicate the negative value of a number.</td>
<td>[Price]-[Discount]</td>
</tr>
<tr>
<td>*</td>
<td>Multiply two numbers.</td>
<td>[Quantity]*[Price]</td>
</tr>
<tr>
<td>/</td>
<td>Divide the first number by the second number.</td>
<td>[Total]/[ItemCount]</td>
</tr>
<tr>
<td>\</td>
<td>Round both numbers to integers, divide the first number by the second number, and then truncate the result to an integer.</td>
<td>[Registered][Rooms]</td>
</tr>
<tr>
<td>Mod</td>
<td>Divide the first number by the second number and return only the remainder.</td>
<td>[Registered] Mod [Rooms]</td>
</tr>
<tr>
<td>^</td>
<td>Raise a number to the power of an exponent.</td>
<td>Number ^ Exponent</td>
</tr>
</tbody>
</table>

**Comparison operators**

You use the comparison operators to compare values and return a result that is either true, false, or Null (an unknown value).

<table>
<thead>
<tr>
<th>Operator</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>Determine if the first value is less than the second value.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Determine if the first value is less than or equal to the second value.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Determine if the first value is greater than the second value.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Determine if the first value is greater than or equal to the second value.</td>
</tr>
<tr>
<td>=</td>
<td>Determine if the first value is equal to the second value.</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Determine if the first value is not equal to the second value.</td>
</tr>
</tbody>
</table>

In all cases, if either the first value or the second value is Null, the result is then also Null. Because Null represents an unknown value, the result of any comparison with Null is also unknown.
Logical operators
You use the logical operators to combine two values and return either a true, false, or Null result. You might also see the logical operators referred to as *Boolean* operators.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>And</td>
<td>Expr1 And Expr2</td>
<td>True when Expr1 and Expr2 are true.</td>
</tr>
<tr>
<td>Or</td>
<td>Expr1 Or Expr2</td>
<td>True when either Expr1 or Expr2 is true.</td>
</tr>
<tr>
<td>Eqv</td>
<td>Expr1 Eqv Expr2</td>
<td>True when both Expr1 and Expr2 are true or both Expr1 and Expr2 are false.</td>
</tr>
<tr>
<td>Not</td>
<td>Not Expr</td>
<td>True when Expr is not true.</td>
</tr>
<tr>
<td>Xor</td>
<td>Expr1 Xor Expr2</td>
<td>True when either Expr1 is true, or Expr2 is true, but not both.</td>
</tr>
</tbody>
</table>

Concatenation operators
You use the concatenation operators to combine two text values into one string.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>string1 &amp; string2</td>
<td>Combines two strings to form one string.</td>
</tr>
<tr>
<td>+</td>
<td>string1 + string2</td>
<td>Combines two strings to form one string and propagates null values.</td>
</tr>
</tbody>
</table>

Special operators
You use the special operators as described in the following table.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Null or Is Not Null</td>
<td>Determines whether a value is Null or Not Null.</td>
</tr>
<tr>
<td>Like &quot;pattern&quot;</td>
<td>Matches string values by using wildcard operators ? and *.</td>
</tr>
<tr>
<td>Between val1 And val2</td>
<td>Determines whether a numeric or date value falls within a range.</td>
</tr>
<tr>
<td>In(string1,string2...)</td>
<td>Determines whether a string value is contained within a set of string values.</td>
</tr>
</tbody>
</table>

Constants
A constant is a known value that does not change and that you can use in an expression. There are four commonly used constants in Access:

- **True**: Indicates something that is logically true.
- **False**: Indicates something that is logically false.
- **Null**: Indicates the lack of a known value.
- **"" (empty string)**: Indicates a value that is known to be empty.

Constants can be used as arguments to a function, and can be used in an expression as part of a criterion. For example, you can use the empty string constant (""") as part of a criterion for a column in a query to evaluate
the field values for that column, by entering the following as the criterion: `<>`"". In this example, `<>` is an operator and "" is a constant. Used together, they indicate that the identifier to which they are applied should be compared to an empty string. The expression that results is true when the identifier's value is anything other than an empty string.

**Note:** Be careful when using the **Null** constant. In most cases, using **Null** in conjunction with a comparison operator will result in an error. If you want to compare a value to **Null** in an expression, use the **Is Null** or the **Is Not Null** operator.

**Examples of expressions**

The following table lists some sample Access expressions and how they are typically used:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>=[RequiredDate]-[ShippedDate]</code></td>
<td>Calculates the difference between the date values in two text box controls (called RequiredDate and ShippedDate) on a report.</td>
</tr>
<tr>
<td><code>Date()</code></td>
<td>Sets the default value for a Date/Time table field to the current date.</td>
</tr>
<tr>
<td><code>ExtendedPrice: CCur([Order Details].Unit Price*[Quantity]*(1-[Discount])/100)*100</code></td>
<td>Creates a calculated field named ExtendedPrice in a query.</td>
</tr>
<tr>
<td><code>Between #1/1/2011# And #12/31/2011#</code></td>
<td>Specifies criteria for a Date/Time field in a query.</td>
</tr>
<tr>
<td><code>=[Orders Subform].Form!OrderSubtotal</code></td>
<td>Returns the value of the OrderSubtotal control on the Orders subform that is on the Orders form.</td>
</tr>
<tr>
<td><code>&gt;0</code></td>
<td>Sets a validation rule for a numeric field in a table — users must enter values greater than zero.</td>
</tr>
</tbody>
</table>

Some expressions begin with the equal (=) operator, and others do not. When you calculate a value for a control on a form or report, you use the `=` operator to start the expression. In other instances, such as when you type an expression in a query or in the **DefaultValue** or **ValidationRule** property of a field or control, you don't use the `=` operator unless you are adding the expression to a Text field in a table. In some cases, such as when you add expressions to queries, Access removes the `=` operator automatically.