**Ereignis Reihenfolge**

Reihenfolge beim Auslösen von Ereignissen
Versionen: Access 2013, 2010, 2007, 2003 und 2002/XP

**FRAGE:** Im VBA-Editor werden in der Liste *Prozedur* die möglichen Ereignisse bei Steuerelementen, Formularen oder Berichten aufgelistet. Es ist aber nicht ersichtlich, in welcher Reihenfolge Ereignisse ausgelöst werden und ziemlich zeitraubend, zum Beispiel ein 'MsgBox "<Ereignisname>"' in alle Ereignisprozeduren einzubauen, um das herauszufinden.
L. Nguyen

**ANTWORT:** Microsoft hat das in der folgenden Techinfo sehr ausführlich dokumentiert:

[**http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx**](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx)

Die Techinfo ist ausgewiesen für Access 2007, aber sie gilt auch für alle älteren und neueren Versionen.

# Order of events for database objects

Show All

Hide All

A single action, such as moving from one control (control: A graphical user interface object, such as a text box, check box, scroll bar, or command button, that lets users control the program. You use controls to display data or choices, perform an action, or make the user interface easier to read.) on an object to another control, can trigger several different events (event: An action recognized by an object, such as a mouse click or key press, for which you can define a response. An event can be caused by a user action or a Visual Basic statement, or it can be triggered by the system.), which occur in a particular sequence. Knowing when events occur and in what order they occur is important, because it can affect how and when your macros (macro: An action or set of actions that you can use to automate tasks.) or event procedures (event procedure: A procedure that is automatically executed in response to an event initiated by the user or program code, or that is triggered by the system.) run. For example, if there are two event procedures that are to be run in a certain order, you want to make sure that the events that they are associated with occur in that same order.

In this article

* [Order of events for controls on forms](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#BM1)
* [Order of events for records on forms](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#BM2)
* [Order of events for forms and subforms](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#BM3)
* [Order of events for keystrokes and mouse clicks](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#BM4)
* [Order of events for reports and report sections](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#BM5)

## Order of events for controls on forms

Events occur for controls on forms when you move the focus (focus: The ability to receive user input through mouse or keyboard actions or the **SetFocus** method. Focus can be set by the user or by the application. The object that has focus is usually indicated by a highlighted caption or title bar.) to a control, and when you change and update data in a control.

**Note**  Microsoft Office Access displays event names slightly differently in the Visual Basic Editor than it does in the property sheet and the **Macro Builder**. For example, the event named **On Got Focus** in the form's property sheet and in the Macro Builder is named **GotFocus** in the Visual Basic Editor. The examples in this article use the Visual Basic for Applications (VBA) format for event names.

### Moving the focus to a control

When you move the focus to a control on a form (for example, by opening a form that contains one or more active controls, or by moving the focus to another control on the same form), the **Enter** and **GotFocus** events occur in this order:

**Enter** **GotFocus**

When you open a form, the **Enter** and **GotFocus** events occur after the events associated with opening the form (such as **Open**, **Activate**, and **Current**), as follows:

**Open** (form) **Activate** (form) **Current** (form) **Enter** (control) **GotFocus** (control)

When the focus leaves a control on a form (for example, when you close a form that contains one or more active controls, or when you move to another control on the same form), the **Exit** and **LostFocus** events occur in this order:

**Exit** **LostFocus**

When you close a form, the **Exit** and **LostFocus** events occur before the events associated with closing the form (such as **Unload**, **Deactivate**, and **Close**), as follows:

**Exit** (control) **LostFocus** (control) **Unload** (form) **Deactivate** (form) **Close** (form)

### Changing and updating data in a control

When you enter or change data in a control on a form and then move the focus to another control, the **BeforeUpdate** and **AfterUpdate** events occur:

**BeforeUpdate** **AfterUpdate**

The **Exit** and **LostFocus** events for a control whose value changes occur after the **BeforeUpdate** and **AfterUpdate** events:

**BeforeUpdate** **AfterUpdate** **Exit** **LostFocus**

When you change the text in a text box or in the text box section of a combo box, the **Change** event occurs. This event occurs whenever the contents of the control change, but before you move the focus to a different control or record (and therefore, before the **BeforeUpdate** and **AfterUpdate** events occur). The following sequence of events occurs for each key that you press in a text box or in the text box section of a combo box:

**KeyDown** **KeyPress** **Dirty** **Change** **KeyUp**

The **NotInList** event occurs after you enter a value in a combo box that is not found in the combo box list and then try to move the focus to another control or record. The **NotInList** event occurs after the keyboard events and the **Change** events for the combo box, but before the events for any other control or the form. If the **LimitToList** property of the combo box is set to **Yes**, the **Error** event for the form occurs immediately after the **NotInList** event:

**KeyDown** **KeyPress** **Dirty** **Change** **KeyUp** **NotInList** **Error**

[Top of Page](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#top)

## Order of events for records on forms

Events occur for records on forms when you move the focus (focus: The ability to receive user input through mouse or keyboard actions or the **SetFocus** method. Focus can be set by the user or by the application. The object that has focus is usually indicated by a highlighted caption or title bar.) to a different record, update data in a record, delete an existing record or records, or create a new record.

### Moving the focus to records and updating data in records

When you move the focus to an existing record on a form, enter or change data in the record, and then move the focus to another record, the following sequence of events occurs for the form:

**Current** (form) **BeforeUpdate** (form) **AfterUpdate** (form) **Current** (form)

When you leave the record whose data has changed, but before you enter the next record, the **Exit** and **LostFocus** events occur for the control that has the focus. These events occur after the **BeforeUpdate** and **AfterUpdate** events for the form, as follows:

**BeforeUpdate** (form) **AfterUpdate** (form) **Exit** (control) **LostFocus** (control) **RecordExit** (form) **Current** (form)

As you move the focus among the controls on a form, events occur for each control. For example, the following sequences of events occur when you do the following:

* Open a form and change data in a control:

**Current** (form) **Enter** (control) **GotFocus** (control) **BeforeUpdate** (control) **AfterUpdate** (control)

* Move the focus to another control:

**Exit** (control1) **LostFocus** (control1) **Enter** (control2) **GotFocus** (control2)

* Move the focus to another record:

**BeforeUpdate** (form) **AfterUpdate** (form) **Exit** (control2) **LostFocus** (control2) **RecordExit** (form) **Current** (form)

### Deleting records

When you delete a record, the following events occur for the form, and Microsoft Office Access displays a dialog box asking you to confirm the deletion:

**Delete** **BeforeDelConfirm** **AfterDelConfirm**

If you cancel the **Delete** event, the **BeforeDelConfirm** and **AfterDelConfirm** events do not occur, and the dialog box is not displayed.

### Creating a new record

When you move the focus to a new (blank) record on a form and then create a new record by typing data in a control, the following sequence of events occurs:

**Current** (form) **Enter** (control) **GotFocus** (control) **BeforeInsert** (form) **AfterInsert** (form)

The **BeforeUpdate** and **AfterUpdate** events for the controls on the form and for the new record occur after the **BeforeInsert** event and before the **AfterInsert** event.

[Top of Page](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#top)

## Order of events for forms and subforms

Events occur for forms when you open or close a form, move between forms, or work with data on a form or subform.

### Opening and closing a form

When you open a form, the following sequence of events occurs for the form:

**Open** **Load** **Resize** **Activate** **Current**

If there are no active controls on the form, the **GotFocus** event occurs for the form after the **Activate** event but before the **Current** event.

When you close a form, the following sequence of events occurs for the form:

**Unload** **Deactivate** **Close**

If there are no active controls on the form, the **LostFocus** event occurs for the form after the **Unload** event but before the **Deactivate** event.

### Moving between forms

When you switch between two open forms, the **Deactivate** event occurs for the first form, and the **Activate** event occurs for the second form:

**Deactivate** (form1) **Activate** (form2)

The **Deactivate** event for a form also occurs when you switch from the form to another object tab in Access. However, the **Deactivate** event does not occur when you switch to a dialog box, to a form whose **PopUp** property is set to **Yes**, or to a window in another program.

**Note**   The **Open** event does not occur if you move the focus to a form that is already open, even if you have moved the focus to that form by performing an **OpenForm** action.

### Working with data on a form

Form and control events occur as you move between records in the form and change data. For example, when you first open a form, the following sequence of events occurs:

**Open** (form) **Load** (form) **Resize** (form) **Activate** (form) **Current** (form) **Enter** (control) **GotFocus** (control)

Similarly, when you close a form, the following sequence of events occurs:

**Exit** (control) **LostFocus** (control) **Unload** (form) **Deactivate** (form) **Close** (form)

If you have changed data in a control, the **BeforeUpdate** and **AfterUpdate** events for both the control and the form occur before the **Exit** event for the control.

### Working with subforms

When you open a form that contains a subform, the subform and its records are loaded before the main form. Thus, the events for the subform and its controls (such as **Open**, **Current**, **Enter**, and **GotFocus**) occur before the events for the form. However, the **Activate** event does not occur for subforms. Therefore, opening a main form triggers an **Activate** event only for the main form.

Similarly, when you close a form that contains a subform, the subform and its records are unloaded after the form. The **Deactivate** event does not occur for subforms. Therefore, closing a main form triggers a **Deactivate** event only for the main form. The events for the controls, form, and subform occur in the following order:

1. Events for the subform's controls (such as **Exit** and **LostFocus**)
2. Events for the form's controls (including the subform control)
3. Events for the form (such as **Deactivate** and **Close**)
4. Events for the subform

**Note**   Because the events for a subform occur after the main form is closed, certain events, such as canceling the closing of the main form from an event in the subform, will not occur. You may need to move these types of validation tests to an event on the main form.

[Top of Page](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#top)

## Order of events for keystrokes and mouse clicks

Keyboard events occur for forms and controls when you press keys or send keystrokes while the form or control has the focus (focus: The ability to receive user input through mouse or keyboard actions or the **SetFocus** method. Focus can be set by the user or by the application. The object that has focus is usually indicated by a highlighted caption or title bar.). Mouse events occur for forms, form sections, and controls on forms when you click the mouse buttons while the mouse pointer is on a form, section, or control. Mouse events also occur when you move the mouse pointer over a form, section, or control.

### Keyboard events

When you press and release a key while a control on a form has the focus (or use the **SendKeys** action or statement (statement: A syntactically complete unit that expresses one specific kind of operation, declaration, or definition. A statement is usually on one line in a procedure or Declarations section, but two or more can be on a line separated by a colon (:).) to send a keystroke), the following sequence of events occurs:

**KeyDown** **KeyPress** **KeyUp**

When you press and release a key or send a keystroke in the ANSI character set (ANSI character set: An 8-bit character set used by Microsoft Windows that allows you to represent up to 256 characters (0 through 255) by using your keyboard. The ASCII character set is a subset of the ANSI set.), the **KeyDown**, **KeyPress**, and **KeyUp** events all occur. If you press and hold down an ANSI key, the **KeyDown** and **KeyPress** events alternate repeatedly (**KeyDown**, **KeyPress**, **KeyDown**, **KeyPress**, and so on) until you release the key; then the **KeyUp** event occurs.

If you press and release a non-ANSI key, the **KeyDown** and **KeyUp** events occur. If you press and hold down a non-ANSI key, the **KeyDown** event occurs repeatedly until you release the key, and then the **KeyUp** event occurs.

If pressing a key triggers another event for a control, that event occurs after the **KeyPress** event but before the **KeyUp** event. For example, if a keystroke changes text in a text box, which triggers a **Change** event, the following sequence of events occurs:

**KeyDown** **KeyPress** **Change** **KeyUp**

If a keystroke causes the focus to move from one control to another control, the **KeyDown** event occurs for the first control, whereas the **KeyPress** and **KeyUp** events occur for the second control. For example, if you change data in a control and then press the TAB key to move to the next control, the following sequences of events occur:

* First control:

**KeyDown** **BeforeUpdate** **AfterUpdate** **Exit** **LostFocus**

* Second control:

**Enter** **GotFocus** **KeyPress** **KeyUp**

### Mouse events

When you click and release a mouse button while the mouse pointer is on a control on a form, the following sequence of events occurs for the control:

**MouseDown** **MouseUp** **Click**

If a control has the focus and you click another control to move the focus to this second control, the following sequences of events occur:

* First control:

**Exit** **LostFocus**

* Second control:

**Enter** **GotFocus** **MouseDown** **MouseUp** **Click**

If you move to another record and then click a control, the **Current** event for the form also occurs before the **Enter** event for the control.

Double-clicking a control causes both the **Click** and **DblClick** events to occur. For example, when you double-click a control other than a command button, the following sequence of events occurs for the control:

**MouseDown** **MouseUp** **Click** **DblClick** **MouseUp**

When you double-click a command button, the preceding sequence of events occurs, followed by a second **Click** event.

The **MouseMove** event for a form, section, or control occurs when you move the mouse pointer over the form, section, or control. This event is independent of the other mouse events.

[Top of Page](http://office.microsoft.com/en-us/access-help/order-of-events-for-database-objects-HA010238988.aspx#top)

## Order of events for reports and report sections

Events occur for reports and report sections when you open a report to print or preview it, or close a report.

### Events for reports

When you open a report to print or preview it and then later close the report or move to another object tab in Access, the following sequence of events occurs for the report:

**Open** **Activate** **Close** **Deactivate**

When you switch between two open reports, the **Deactivate** event occurs for the first report, and the **Activate** event occurs for the second report:

**Deactivate** (report1) **Activate** (report2)

The **Deactivate** event for a report also occurs when you switch from the report to another object tab in Access. However, the **Deactivate** event does not occur when you switch to a dialog box, to a form whose **PopUp** property is set to **Yes**, or to a window in another program.

When you open a report that is based on a query, Access triggers the **Open** event for the report before it runs the underlying query. As a result, you can set the criteria for the report by using a macro or event procedure that responds to the **Open** event. For example, the macro or event procedure can open a custom dialog box in which you enter report criteria.

### Events for report sections

When you print or preview a report, the **Format** and **Print** events occur for the report sections after the **Open** and **Activate** events for the report and before the report's **Close** or **Deactivate** events:

**Open** (report) **Activate** (report) **Format** (report section) **Print** (report section) **Close** (report) **Deactivate** (report)

In addition, the following events can occur during or after formatting but before the **Print** event:

* The **Retreat** event occurs when Access returns to a previous section during report formatting.
* The **NoData** event occurs if there are no records displayed by the report.
* The **Page** event occurs after formatting but before printing. You can use this event to customize the appearance of the printed report.