

SetOrderBy

When you run the *SetOrderBy* action, the sort is applied to the table, form or datasheet (for example, query result) that is active and has the focus.

tags:
Actions

Applies to ...

Object	Description
DoCmd	Root class. When present, its name must be "DoCmd" but the object name is optional.

Syntax

`[DoCmd.]SetOrderBy([orderby], [controlname])`

Argument	Optional	Type	Description
<code>orderby</code>	Yes	String	The name of the field or fields on which you want to sort records. When you use more than one field name, separate the names with a comma (,).
<code>controlname</code>			The name of a subform of the active form.

Remarks

- The *SetOrderBy* action must not be called from a **standalone form**.
- The *SetOrderBy* action is applied on the current window. To make a window current, use the **SelectObject** action. If the current window is neither a form, a table datasheet or a query datasheet, the *SetOrderBy* action returns **False** and ignores the request.
- The *SetOrderBy* action, when applied to a table or a query datasheet, does NOT work in OpenOffice (LibreOffice OK).
- To sort a table or a query with *SetOrderBy*, the table or the query must be open. Eventually use therefore the **OpenTable** or **OpenQuery** actions.
- If *controlname* is present, the active window is expected to be a form. Otherwise *SetOrderBy* returns **False** and the request is ignored. The *controlname* is NOT case-sensitive. If *controlname* does not exist in the active form or is not the name of one of its **subforms** then the action generates a run-time error.
- When records are sorted with *SetOrderBy* the first record (if it exists ...) becomes the current record.
- Once applied, the sorting sequence is preserved for subsequent table or query openings during the same *LibreOffice/OpenOffice* session. It will become persistent when the database file (*.odb) is saved.
- In the orderby argument, record and field names may be surrounded by square brackets. They will be replaced with the correct character surrounding such names in SQL statements targeted to be run on the concerned RDBMS (Relational Database Management System).
- To sort records in descending order, type DESC immediately after the field that needs to be sorted in descending order.
- Giving the null-length string ("") as orderby argument resets any pre-existing sorting sequence.

Error messages

Argument nr.X [Value = '...'] is invalid
Control '...' not found in parent (form, grid or dialog) '...'
Subform '...' not found in parent form '...'

See also

ApplyFilter
Filter

FilterOn
GoToRecord
SelectObject
SetFocus

Example

Sort the records of a subform

```
Dim sSetOrderBy As String
Const cstFormName = "INVOICE"
Const cstSubFormName = "SubForm"
    OpenForm(cstFormName)
    sSetOrderBy = "[PURCHASE_PRICE] DESC"
    SelectObject(acForm, cstFormName)
    DoCmd.SetOrderBy(sSetOrderBy, cstSubFormName)
```

Bookmark this page » » [SetOrderBy](#)