

[Home](#) x [Database Functions](#) x [Definitions](#) x [Error handling](#) x [Events handling](#) x [Methods](#) x
[Find](#) x [GetChunk](#) x [Objects](#) x [Module](#) x [Properties](#) x [On \(control event\)](#) x
[On \(database event\)](#) x [On \(dialog event\)](#) x [On \(form event\)](#) x [ProcBodyLine](#) x [ProcCountLines](#) x
[ProcOfLine](#) x [ProcStartLine](#) x [Shortcuts](#) x [Snippets of Code](#) x [Module example](#) x
[On control event example](#) x [On dialog event example](#) x [On form event example](#) x [Special controls](#) x
[Tutorial](#) x [DLookupSamples](#) x [Object Model](#) x [Standalone Forms](#) x [User's Guide](#) x

Standalone Forms

S *andalone forms* are stored in non-Base AOO/LibO documents (Writer, Calc, ...), usually (but not necessarily) built from a normal *form* Base object. These forms and controls may be linked with data located in databases by referring to a Base document (".odb" file) in the data tab of the form properties.

tags:
[Menu](#)

Each such form may access a separate database document !

To make such forms, documentation is available on several forums, e.g. [here](#).

Access2Base may be invoked from macros stored in documents containing one or more *standalone forms*. However NOT ALL the properties, methods, actions etc. are meaningful in such a context. Find below the status by function. If not mentioned a function is available without restriction.

In addition a *standalone form* does see the *Table* and *Query* objects stored within the database document (".odb" file) where it is connected with.

At the opposite:

A *standalone form* **CANNOT INTERACT** with (either standalone or embedded) forms stored in **OTHER DOCUMENTS**.
 If it is necessary to exchange data between individual *LibO/AOO* applications, use **TempVar** objects. Their scope is the whole *LibO/AOO* session.

A *standalone form* can **READ and MODIFY DATA** in its related database, can **READ its DESIGN** but cannot modify it.

Several documents containing standalone forms might be opened at the same time, all using the *Access2Base* API.

Use the Access2Base library

To be able to invoke the *Access2Base* API from a non-Base AOO/LibO document you have to

- Have a minimal knowledge of the Basic IDE and of the Basic programming language.
- Open the targeted document and edit at least next macro:

Connect all standalone form contained in a non-Base document to their target databases:

```
Sub DBOpen(Optional poEvent As Object)
    If GlobalScope.BasicLibraries.hasByName("Access2Base") then
        GlobalScope.BasicLibraries.loadLibrary("Access2Base")
    End If
    Call Application.OpenConnection(ThisComponent)
End Sub
```

- Assign with menu items **Tools + Customize...** (**Events** tab) the above Sub ("DBOpen" in the example but use the name of your choice) to the *OpenDocument* event. Save in the document file itself.
- Optionally associate next code with the *"View is going to be closed"* document event.

```
Sub DBClose(Optional poEvent As Object)
    Call CloseConnection()
End Sub
```

- *Close* and re-open the file. This will trigger the *OpenDocument* event.
- Start programming macro's. Associate them with *form* or *control* events if relevant

Next functions are available with limitations

Function	Limitation
AllForms	The <i>AllForms</i> collection is limited to the <i>standalone forms</i> stored in the same document. When a document contains one single form - which is very often the case - the smart syntax for invoking that single form is simply: <code>AllForms(0)</code> which will return a <i>form object</i> representing the single <i>standalone form</i> itself, giving subsequently access to its individual controls.
FindRecord	If the form contains subforms and if the targeted field to search on is in that subform, then the <i>OnlyCurrentField</i> argument must be a <i>string</i> containing a shortcut if the targeted grid or grid column is in the subform.
GoToRecord	The <i>ObjectType</i> argument must be equal to <code>acDataForm</code> and the <i>ObjectName</i> argument must be a <i>string</i> containing either the name of the targeted form or a shortcut to it.
SelectObject	The objects opened from a Base document (".odb") - tables, queries, etc. - can be selected from a <i>standalone form</i> . Similarly <i>Standalone forms</i> can be selected from a Base document (".odb") via their filenames. However no interaction is possible between them thru the <i>Access2Base</i> API. As an example it is not possible to know from the current <i>standalone form</i> the content of a <i>control</i> located in another <i>database form</i> or a <i>standalone form</i> belonging to another document.
Height Width Move	These methods, even when applied to a specific form, act on the containing window, i.e. on all its forms.
OpenArgs	The <i>OpenArg</i> (read-only) property always returns a zero-length string.

Next functions are not available

Access2Base will generate an error (or sometimes ignore the request) if one of next functions is invoked from a *standalone form* context.

Function	Comment
ApplyFilter	Use the Filter and FilterOn form properties instead.
Close	Forbidden when applied to a form or another database object. Closing a Recordset is allowed.
CopyObject	A standalone form has limited access to database objects. The offered facilities are mainly recordset manipulations thru tables and queries.
OpenForm, OpenQuery, OpenReport, OpenTable	
CreateQueryDef	
CreateTableDef, CreateField, Add	
Delete (table-query)	
GoToControl	Use the SetFocus method instead.
Quit	
ShowAllRecords	Use the FilterOn and/or OrderByOn form properties instead.

See also ...

CloseConnection
OpenConnection

Bookmark this page » » [Standalone Forms](#)