Home x Install x Object Model x Tutorial x Standalone Forms x User's Guide x
Why Access2Base ? x Actions x ApplyFilter x Close x CopyObject x FindNext x FindRecord x
GetHiddenAttribute x GoToControl x GoToRecord x Maximize x Minimize x MoveSize x
OpenForm x OpenQuery x OpenReport x OpenSQL x OpenTable x OutputTo x Quit x
RunApp x RunCommand x RunSQL x SelectObject x SendObject x SetHiddenAttribute x
SetOrderBy x ShowAllRecords x SysCmd x AllDialogs x Collections x

AllDialogs

The *AllDialogs* collection describes instances of all **dialogs** present in the currently loaded dialog libraries.

tags:
Collections

Applies to ...

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

Syntax

```
[Application.]AllDialogs()
[Application.]AllDialogs(index)
[Application.]AllDialogs(dialogname)
```

Argument #1	Type	Returned value
	absent	A **Collection** object
index	integer long	A **Dialog** object corresponding to the index-th item in the AllDialogs() collection. The 1st dialog is AllDialogs(0), the 2nd is AllDialogs(1) and so on ... The last one is AllDialogs.Count - 1.
dialogname	string	A **Dialog** object having the argument as name. The argument is NOT case-sensitive.

Remarks

- *Access2Base* will scan first the dialogs present in the current Base document (".odb" file) or current non-Base document containing one or more **standalone forms** (".odt", ".ods", ... file) and continue the search thru all currently loaded libraries. The *Access2Base* library itself however will be skipped.
- The *dialogname* argument is not case sensitive.
- Homonyms within the scanned libraries should be avoided. Only their non case-sensitive name can differentiate them.

Error messages

Argument nr. 1 [Value = '...'] is invalid
Out of array range or incorrect array size for collection AllDialogs()
Dialog '...' not found in the currently loaded libraries

See also ...

EndExecute
Execute
Start
Terminate

Examples

Display a dialog

```
Dim oDialog As Object, lExecute As Long
Const dlgOK = 1
Const dlgCancel = 0
```

```
        oDialog = Application.AllDialogs("myDialog")
        oDialog.Start
        lExecute = oDialog.Execute
        Select Case lExecute
                Case dlgCancel                  '       Cancel button pressed
                        '       ... do probably nothing ...
                Case dlgOK                      '       OK button pressed
                        '       ... process the dialog, all controls are still available
                Case Else                       '       Dialog interrupted programmatically
                        '       ... process the dialog based on the returned value
        End Select
        oDialog.Terminate
```

Bookmark this page » » AllDialogs