

[Home](#) x [Install](#) x [Full Index](#) x [Tutorial](#) x [EnumerateControls](#) x [FindOutTableExists](#) x [UseVariablesInSQL](#) x [CreateRecordsetFrom](#) x [AddRecordToRecordset](#) x [CountRecordsRecordset](#) x [LimitsRecordset](#) x [MixAccess2baseAndUNO](#) ! [User's Guide](#) x [AllForms](#) x [DatabaseWindow](#) x [ShortcutNotationMore](#) x [DLookupSamples](#) x [CalculatedField](#) x [MultiSelectListBoxSelectForm](#) x [FillAutoControlValue](#) x [CarryToNewRecord](#) x [BrowseThruControls](#) x [TipTextForLongValues](#) x [AskBeforeSaving](#) x [Sync2Combos](#) x [ZoomOnImage](#) x [AddAllToBox](#) x [KeepFormsSynchro](#) x [SelectListBoxOnFirstLetters](#) x [MoveItemsBetweenListboxes](#) x [SimulateTabbed](#) x [SearchStandalone](#) x [CalculatorDialog](#) x [ExploreTables](#) x [ExtractDataTable](#) x [FindPositionRecordset](#) x [DMedian function](#) x [DPercentile](#) x [ImportImages](#) x [ExportImages](#) x [CrossTabQuery](#) x [DbaccessFromCalc](#) x [Standalone Forms](#) x [Add](#) x [AddItem](#) x [AddNew](#) x [CancelUpdate](#) x [Clone](#) x [Close \(method\)](#) x [CloseAllRecordsets](#) x [CloseConnection](#) x [CreateField](#) x [CreateQueryDef](#) x [CreateTableDef](#) x [CurrentDb](#) x [Delete](#) x [Delete \(table-query\)](#) x [Edit](#) x [EndExecute](#) x [Execute \(commandbarcontrol\)](#) x [Execute \(dialog\)](#) x [Execute \(query\)](#) x [getProperty](#) x [GetRows](#) x [hasProperty](#) x [Move](#) x [Move \(recordset\)](#) x [OpenConnection](#) x [OpenDatabase](#) x [OpenRecordset](#) x [OptionGroup \(Method\)](#) x [ReadAllBytes](#) x [ReadAllText](#) x [Refresh](#) x [Remove](#) x [RemoveAll](#) x [RemoveItem](#) x [Requery](#) x [Reset](#) x [RunSQL](#) x [SetFocus](#) x [setProperty](#) x [Start](#) x [Terminate](#) x [Update](#) x [WriteAllBytes](#) x [WriteAllText](#) x [Methods](#) x [Objects](#) x [Application](#) x [CommandBar](#) x [CommandBarControl](#) x [Control](#) x [Database](#) x [Dialog](#) x

Dialog

A *Dialog* **object** describes a dialog located in one of the accessible Basic libraries. If the property **IsLoaded** returns *True* then the dialog is active.

tags:
Objects

Functions returning a dialog object

Parent object	Function	Type	Argument	Description
Application	AllDialogs	Collection	Integer or String	<code>Application.AllDialogs("myDialog")</code> returns an object corresponding with the <code>myDialog</code> dialog
	getObject		String	<code>getObject("Dialogs!myDialog")</code> returns an object corresponding with the <code>myDialog</code> dialog. <code>myDialog</code> must be active.

Properties

Property	Type	Read only	Description or UNO object
Caption			Specifies the text that appears in the title bar.
Height			Specifies the height of the dialog.
IsLoaded		Y	True if dialog is active.
Name		Y	Specifies the real name of the dialog
ObjectType		Y	Returns "DIALOG"
Visible			Shows or hides the dialog without making it inactive.
Width			Specifies the width of the dialog.
UnoDialog	UNO	Y	com.sun.star.awt.XControl The output of the <i>CreateUnoDialog</i> Basic function.

Methods

Method	Argument(s)	Return	Description
EndExecute	returnvalue	Long	Stop the execution of the dialog and return <i>returnvalue</i> .
Execute			Display the dialog and interact with the user.
getProperty	property	Variant	Return the value of the given property.
hasProperty	property	Boolean	Return True if the Dialog has the given property.

Move	coordinates		Return True if Dialog has been moved successfully.
setProperty	property value		Return True if the given property could be set.
Start			Initialize the dialog.
Terminate			Close all processing concerning the actual dialog.

Remarks

Each *Dialog* **object** has a Controls **collection**, that contains all controls on the dialog. You can refer to a specific control on a dialog by referring to the **Controls** collection.

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 » » [Dialog](#)