Application

T he *Application* class is the root class from which the initial **collections** are derived.
The *Application* class must not be instantiated. The only preset instance MUST be called
`Application`. In fact it is implemented simply as a Basic module.

tags:
Objects

As an example, next statements are equivalent :

```
        Set ofForm = Forms("anyform")
```

and

```
        Set ofForm = Application.Forms("anyform")
```

However, <u>the second one is recommended</u> to avoid risks of naming collisions between concurrent Basic libraries.

Functions returning objects

Function	Description
AllDialogs	Collection of dialogs, active or not.
AllForms	Collection of forms, open or not.
CommandBars	Collection of tool-, menu- and statusbars associated with the current window.
CurrentDb	Returns the current **database** object, i.e. the only database from a Base (".odb" document, or the database linked with the first **standalone form** contained in a non-Base (Writer, Calc, ...) document.
Events	Returns the currently running **event** as an object.
Forms	Collection of open forms.
OpenDatabase	Returns a **database** object giving access to the data stored in that database.
TempVars	Collection of temporary variables available thru the whole LibreOffice/OpenOffice session.

Properties

Property	Type	Description
CurrentUser	String	The logon name of the current user.
DAvg **DCount** **DLookup** **DMin, DMax** **DStDev, DStDevP**	Any	The database functions.

DSum **DVar, DVarP**		
ProductCode	String	Returns "Access2Base x.y.z" where x.y.z equals the version number of the library.
Version	String	Returns "LibreOffice x.y.z" or "OpenOffice x.y.z" where x.y.z is the version number.

Methods

OpenConnection	To be run to connect the application to the database.

Bookmark this page » » Application