DBAppMon v1.2 Sep 95

Description DBAppMon notifies a Visual Basic program every time an application or a DLL starts or exits. It also contains properties for retrieving various information about active tasks and modules as well as file version info.

File Name DBAPPMON.VBX

Object Type DBAppMon

Remarks DBAppMon is a VB interface to the libraries TOOLHELP.DLL and VER.DLL. By installing a Toolhelp notification callback, DBAppMon is able to supply a VB program with information regarding application and module startup and termination. The primary reason DBAppMon was written was the need to wait for an application and all its children to terminate. In the process, most of WPS' (a program supplied with the CDK) functionality was included.

By being a little creative and combining DBAppMon's features with more standard API calls like *GetWindow* and *GetWindowInstance*, it's possible find out a whole lot about the currently running applications. Ever wanted to use the VB *AppActivate* command, but found that the application in question changes its caption on the fly? DBAppMon is your rescue!

About This control was developed by Dan Byström. For more information, contact me at e-mail: "dan.bystrom@visual-design.se" or phone: +46 708 68 65 78 (no support calls, please). I would be happy to discuss development of customized VBX'es or OCX'es for you.

Distribution You have the right to do whatever you want with DBAppMon, *as long as you don't attempt to modify any of its code*. "Do whatever you want" includes using DBAppMon in your own commercial applications and distributing it for free.

When the control is loaded in design mode a message is sometimes displayed. *This message may not be removed or changed in any way*. Anyway, the message won't ever appear at run-time.

I'm giving DBAppMon away for free. If you should decide to use it in an application of yours, this means that I have saved you a whole lot of trouble, time and $$$ doing it yourself. Therefore I think a nice gesture would be to include some sort of credit in your application's about box and/or documentation. You could include the name of the VBX, my name and my e-mail address. I would be glad if you did this. Anyway - happy VB programming with DBAppMon!

Revision history

Jan 95: Beta release.
- v0.9: New property added: *MyTask.*
- v1.0: More resistant to crashing applications.
  The *Monitor* property doesn't show up in design mode. It never should.
- v1.1: Version info had problem with different language versions.
- v1.2: New property added: *HotKey*.
  New event added: *HotKey*.

Properties

*AllModules	*AllTasks	*HotKey	Index
*ModuleFileName	*ModuleFindName	*ModuleLookupName	*ModuleName

*ModuleUsage	*Monitor	*MyTask	Name
Parent	Tag	*TaskFileName	*TaskInstance
*TaskModule	*TaskName	*TaskParent	*VerLanguageName
*VerLanguages	*VerQueryValue	*VerReadInfo	

* = The property applies only to DBAppMon.

Events

*DLLExit	*DLLStart	*AppExit	*AppStart
*HotKey			

* = The event applies only to DBAppMon.

AllModules Property

Description Retrieves all currrently active modules in the system.

Usage *DBAppMon*.**AllModules**

Remarks The property returns all module handles in a comma separated string.

Data Type String

AllTasks Property

Description Retrieves all currrently active tasks in the system.

Usage *DBAppMon*.**AllTasks**

Remarks The property returns all task handles in a comma separated string.

Data Type String

HotKey Property

Description The HotKey property array maintains a list of up to 10 hotkey combinations. Each hotkey will fire the HotKey event when the keyboard combination gets pressed, regardless of what application is currently active (as long as it's not a 32-bit application).

Usage *DBAppMon*.**HotKey**(*Index*) [= *KeyCode%*]

Remarks The key code value should be calculated like this: take the virtual key code (like in the VB *KeyDown* event) and then add a combination of the following constants:

&H0100&	shift key pressed simultaneously
&H0200&	ctrl key pressed simultaneously
&H0400&	alt key pressed simultaneously
&H0800&	don't pass the key combination to the active application

Data Type Integer

ModuleFileName Property

Description Retrieves the file name from a module handle.

Usage *DBAppMon*.**ModuleFileName**(*hModule*)

Data Type String

ModuleFindName Property

Description Retrieves the module handle of a module name. The module name used for the search

is the content of the property *ModuleLookupName*.

Usage *DBAppMon*.**ModuleFindName**

Data Type Integer

ModuleLookupName Property

Description Gets or sets the module name used in subsequent calls to *ModuleFindName*. No action is performed until *ModuleFindName* is called.

Usage *DBAppMon*.**ModuleLookupName** = [*modulename$*]

Data Type String

ModuleName Property

Description Retrieves the module name from a module handle.

Usage *DBAppMon*.**ModuleName**(*hModule*)

Data Type String

ModuleUsage Property

Description Retrieves a module's usage count from a module handle.

Usage *DBAppMon*.**ModuleUsage**(*hModule*)

Data Type Integer

Monitor Property

Description Starts or stops the notification events (*DLLStart*, *DLLExit*, *AppStart* and *AppExit*). Note that these events won't be fired for 32-bit applications.

Usage *DBAppMon*.**Monitor** = [*setting%*]

Data Type Integer (Boolean)

MyTask Property

Description Retrieves the task handle of the application itself.

Usage *DBAppMon*.**MyTask**

Remarks This property just calls the API function *GetCurrentTask()*.

Data Type Integer

TaskFileName Property

Description Retrieves the file name from a task handle.

Usage *DBAppMon*.**TaskFileName**(*hModule*)

Data Type String

TaskInstance Property

Description Retrieves the task instance handle from a task handle.

Usage *DBAppMon*.**TaskInstance**(*hModule*)

Remarks This is the same handle as returned by the VB *Shell* function.

Data Type Integer

TaskModule Property

Description Retrieves the module handle from a task handle.

Usage *DBAppMon*.**TaskModule**(*hModule*)

Data Type Integer

TaskName Property

Description Retrieves the task name from a task handle.

Usage *DBAppMon*.**TaskName**(*hModule*)

Data Type String

TaskParent Property

Description Retrieves the task's parent from a task handle.

Usage *DBAppMon*.**TaskParent**(*hModule*)

Remarks This is the task handle of the application which launched the task.

Data Type Integer

VerLanguageName Property

Description Retrieves a language name of version info from a file. A file may contain multiple languages.

Usage *DBAppMon*.**VerLanguageName** (*Language%*)

Remarks Legal language numbers range from zero to *VerLanguages*-1. After reading this property, *Language%* becomes the *current language* used when retrieving version fields with *VerQueryValue*. This property retrieves a string in the format "LLLLCCCC Language name", where the first 4 characters consists of the language code number in hex, characters 5 to 8 are the code page number in hex, character 9 is always a space and the remaining characters, starting at the 10th position, are the language name in readable text.

Data Type String

VerLanguages Property

Description Retrieves the number of languages the version info is available in.

Usage *DBAppMon*.**VerLanguages**

Remarks After version info has been read from a file into memory, this propery contains the number of languages the versio info is available in.

Data Type Integer

VerQueryValue Property

Description Retrieves selected version information previously read into memory with the *VerReadInfo* property.

Usage *DBAppMon*.**VerQueryValue** = *filedname$*
DBAppMon.**VerQueryValue**

Remarks This property serves double duty. When written, it stores the name of a version field, and when read, it fetches the value of that particular field. Some common field names are: "Comments", "CompanyName", "FileDescription", "FileVersion", "InternalName", "LegalCopyright", "LegalTrademarks", "OriginalFilename", "PrivateBuild", "ProductName", "ProductVersion", and "SpecialBuild". Note: If the version info is available in multiple languages, the language last read through the *VerLanguageName* property is used.

Data Type String

Example

```
DBAppMon1.VerReadInfo = "c:\windows\system\dbappmon.vbx"
      DBAppMon1.VerQueryValue = "CompanyName"
      MsgBox DBAppMon1.VerQueryValue
      DBAppMon1.VerReadInfo = ""
```

VerReadInfo Property

Description Retrieves version info from a file.

Usage *DBAppMon*.**VerReadInfo** = *filename$*

Remarks The version info is read into memory and kept there for further investigation through the *VerQueryValue* property. If the file doesn't exist or if it doesn't contain verion info, a trappable error 52 (Bad file name or number) is generated. To free the few bytes used to hold the version info, set this property to an empty string.

Data Type String

DLLExit Event

Description Occurs after a DLL has been unloaded.

Sub *DBAppMon*_**DLLExit** (*hModule* **As Integer**)

Remarks Since the event occurs *after* the DLL has unloaded, *hModule* has is invalid. It shall only be used to be compared with previously stored hModules.

DLLStart Event

Description Occurs after a DLL has been loaded.

Sub *DBAppMon*_**DLLStart** (*hModule* **As Integer**)

Remarks *hModule* may be stored for later use or passed to any of the *ModuleFileName*, *ModuleName* or *ModuleUsage* properties.

AppExit Event

Description Occurs after an application has terminated.

Sub *DBAppMon*_**AppExit** (*hTask* **As Integer**, *nExitCode* **As Integer**)

Remarks Since the event occurs *after* the application has terminated, *hTask* is invalid. It shall only be used to be compared with previously stored hTasks. *nExitCode* contains the application's exit code (*ErrorLevel* for DOS applications).

AppStart Event

Description Occurs after an application has started.

Sub *DBAppMon*_**AppStart** (*hTask* **As Integer**)

Remarks *hTask* may be stored for later use or passed to any of the *TaskFileName*, *TaskInstance*,

TaskModule, *TaskName* or *TaskParent* properties.

HotKey Event

Description Occurs when a hot key combination, specified with the *HotKey* property, has been pressed.

Sub *DBAppMon*_**HotKey** (*KeyCode* **As Integer**, *KeyInfo* **As Integer**)

Remarks *KeyCode* is the same key code as previously passed to the *HotKey* property and *KeyInfo* may be interpreted like this:

bit 0-7:	keyboard scan code
bit 8:	extended key
bit 9-10:	not used
bit 11-12:	used internally by Windows
bit 13:	context code
bit 14:	previous key state
bit 15:	key transition state

Version control

The following code shows an easy way to check the version of DBAPPMON.VBX before it is accessed by VB. In a global module, put the following declaration:

```
Declare Function DBAppMonVersion Lib "dbappmon.vbx" () As Integer
```

Then use a Sub Main() as your program's entry point:

```
Sub Main()
 If DBAppMonVersion() < &H120& Then
   MsgBox "Your DBAAPPMON.VBX is too old for this program!", 16
   End
 End If
'Load your main form here
End Sub
```