

FileLockAnalyzer – Documentation

This is the documentation for the Access Dev Tools FileLockAnalyzer class module.

System Requirements

This component requires:

- Windows 7 / Windows Server 2008, or newer
- Microsoft Office 2010 (VBA 7), or newer

FileLockAnalyzer supports the 32bit and 64bit editions of Microsoft Office.

Integration

To use FileLockAnalyzer in your own VBA application...

1. Open your project in the Host Application (Microsoft Access, Microsoft Excel, Microsoft Word etc.).
2. Then open the VBA IDE for your project.
3. Use the Menu “Debug” – “Compile” to make sure the project compiles. – If there are any compile errors, it’s recommended you fix them before you integrate the FileLockAnalyzer components into your project.
4. Use the Menu “File” – “Import File” to import the file “FileLockAnalyzer.cls” and “LockProcessInfo.cls” into your project.
5. Finally use the Menu “Debug” – “Compile” again to make sure the project still compiles.

The integration is now complete, and you can use FileLockAnalyzer in your VBA code.

Usage

The class FileLockAnalyzer will be automatically available with a default instance. You do not need to create an instance of the FileLockAnalyzer class to use it.

Here is a very simple example showing how to retrieve the name of the first application locking a file:

```
Public Sub SimpleExample()  
  
    Dim lockingApplication As String  
    lockingApplication = _  
        FileLockAnalyzer.GetFirstLockingApplicationForFile("c:\tmp\myworddoc.docx")  
    If Len(lockingApplication) > 0 Then  
        MsgBox "The file is currently locked by: " & lockingApplication  
    Else  
        MsgBox "The file is currently not locked by any application."  
    End If  
  
End Sub
```

For more sophisticated examples, please look at the code in the FileLockAnalyzerDemo.accdb file, which is included in the download package.

FileLockAnalyzer class

The FileLockAnalyzer class is the primary component for interaction and contains most of functionality of this component.

Methods

[GetFirstLockingApplicationForFile](#)

The `GetFirstLockingApplicationForFile` will retrieve the name of an application locking a file.

```
Public Function GetFirstLockingApplicationForFile(ByVal FullFileName As String) As String
```

Arguments

FullFileName must be the full path to the file you want to retrieve lock information for. This can be a file on the local computer or a file on a mapped network share.

Note: For a file on a network share `FileLockAnalyzer` can only find lock information for other applications **on the same computer** that are locking this file.

Return Value

The return value is a string containing the application name of the first application, which is holding a lock on the file.

[GetLocksForFile](#)

The `GetLocksForFile` function will retrieve all available file lock information for a single file.

```
Public Function GetLocksForFile(ByVal FullFileName As String) As Collection
```

Arguments

FullFileName must be the full path to the file you want to retrieve lock information for. This can be a file on the local computer or a file on a mapped network share.

Note: For a file on a network share `FileLockAnalyzer` can only find lock information for other applications **on the same computer** that are locking this file.

Return value

The function returns a `Collection` of `LockProcessInfo` objects. There will be one `LockProcessInfo` for each `Application` that is locking the file.

If there are no file locks found, the collection will be empty. Always check the `Count` property of the collection before accessing items in the collection.

[LockProcessInfo class](#)

The `LockProcessInfo` class is a “data bag” class to return the information about the locking processes.

Properties

[ApplicationType](#)

The application type of the locking process. See the `AppType` enum for more information.

```
Public ApplicationType As AppType
```

[AppName](#)

`AppName` is the name of the application holding the lock. This is the “friendly name” of the application, not the executable name.

```
Public AppName As String
```

[AppTypeText](#)

`AppTypeText` returns a text representation of the `ApplicationType` property.

```
Public Property Get AppTypeText() As String
```

AppStatus

AppStatus is a combination of values from the [RM_APP_STATUS enumeration](#).

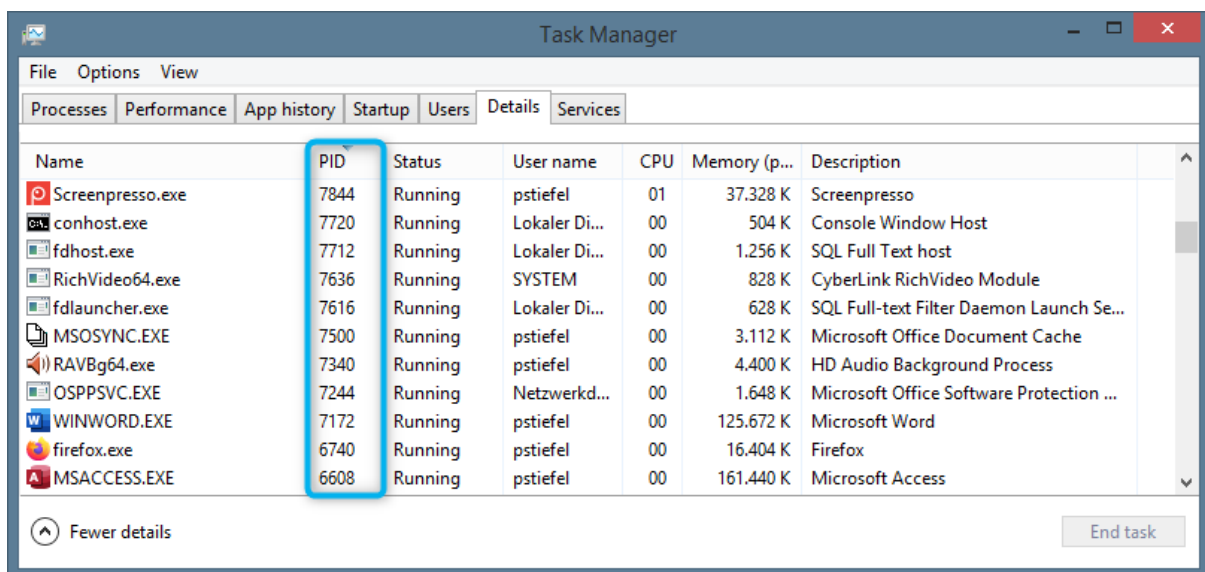
Note: This is included for completeness only and probably does not provide any value in the context of an VBA application.

Public AppStatus As Long

ProcessId

This is the process id of the process holding the lock. The value can be used for further automated interaction with the process.

Note: The ProcessId is displayed in the PID column in the Details view of Windows Task Manager.



Name	PID	Status	User name	CPU	Memory (p...	Description
Screenpresso.exe	7844	Running	pstiefel	01	37.328 K	Screenpresso
conhost.exe	7720	Running	Lokaler Di...	00	504 K	Console Window Host
fdhost.exe	7712	Running	Lokaler Di...	00	1.256 K	SQL Full Text host
RichVideo64.exe	7636	Running	SYSTEM	00	828 K	CyberLink RichVideo Module
fdlauncher.exe	7616	Running	Lokaler Di...	00	628 K	SQL Full-text Filter Daemon Launch Se...
MSOSYNC.EXE	7500	Running	pstiefel	00	3.112 K	Microsoft Office Document Cache
RAVBg64.exe	7340	Running	pstiefel	00	4.400 K	HD Audio Background Process
OSPPSVC.EXE	7244	Running	Netzwerkd...	00	1.648 K	Microsoft Office Software Protection ...
WINWORD.EXE	7172	Running	pstiefel	00	125.672 K	Microsoft Word
firefox.exe	6740	Running	pstiefel	00	16.404 K	Firefox
MSACCESS.EXE	6608	Running	pstiefel	00	161.440 K	Microsoft Access

Public ProcessId As Long

ProcessStartTime

The start time of the process holding the lock.

Note: This is not necessarily the time the locked file was opened in the application.

Public ProcessStartTime As Date

ServiceName

This is the service name if the locking process is part of a Windows service.

Public ServiceName As String

TSSessionId

The terminal session id the process is running in.

Note: TSSessionId will be 1 for all processes running in the context of a normal desktop session. For connections to Windows via Remote Desktop (RDP), including connections to a Windows in a virtual machine, the value will be > 1.

Public TSSessionId As Long

Enums

AppType

The AppType enum is a clone of the [RM_APP_TYPE enumeration](#) of the Restart Manager API. Please see the Microsoft's documentation for further information.

Note: The documentation of RM_APP_TYPE enumeration mentions for the RmCritical value: *"The process may belong to the primary installer that started the Restart Manager."* In the FileLockAnalyzer implementation, your VBA application started the Restart Manager session and thus will be classified as "Critical" application if it is included in the lock results.