WinInfo XObject Documentation

What is WinInfo?

Windows configurations can be quite different. The WinInfo XObject gives you easy access to user-defined, customized and system-level Windows information. WinInfo also provides easy read and write functions to Windows initialization (*.INI) files. WinInfo is optimized for Microsoft Windows 3.x 16-bit use.

Why use WinInfo?

You want your applications to be as friendly as possible and to perform flawlessly. At the same time, you may need to on-the-fly configure the playback PC. WinInfo allows you to retrieve system settings and later reset them. In addition, certain Windows settings beyond your control may have adverse effect on your application's performance. WinInfo allows you to access information about a Windows playback environment and alter these settings or alert your users to changes they may want to make to their system. It may also be used as a custom support tool, providing easy access to the end-user's unique Windows configuration.

What's in the movie?

The projector movie WININFO.EXE hilights the abilities of the WinInfo Xobject: what each method does, optional parameters to use with each method, what information the method returns, and sample Lingo scripts for each method. The WinInfo Sampler uses the WinInfo XObject to provide you with a configuration readout of the system playing back this file.

Who created WinInfo?

WinInfo was developed by **createMedia**, a Macromedia Authorized Developer since 1991, specializing in Windows and cross-platform multimedia application design and development. **createMedia**'s principle software designer, Kenn Johnson, wrote the original Windows Player 1.0 and 1.1 User Guide and Guide to Cross-Platform Development manuals and was MacroMinds's Lead QA Engineer for the original Windows Player 1.0.

createMedia provides Windows Xobject development services, as well as Macromedia Director, Microsoft Visual Basic and Asymetrix Toolbook design and development.

WinInfo Methods

mNew

What it does

mNew creates a new instance of the WinInfo Xobject

Parameters

none

Lingo Examples

```
openXLib "WININFO.DLL"
set obj = WININFO(mNew)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

mNew returns	1 if a new object was created; 0 if no object was created.

mDispose

What it does

mDispose destroys an existing instance of the WinInfo Xobject

Parameters

none

Lingo Examples

```
openXLib "WININFO.DLL"
set obj = WININFO(mNew)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

mDispose returns nothing

mName

What it does

mName returns the name of the WinInfo Xobject, "WININFO"

Parameters

none

Lingo Examples

```
openXLib "WININFO.DLL"
put wininfo(mName)
closeXLib "WININFO.DLL"
```

What it returns

mName returns the name of the WinInfo Xobject, "WININFO"

mDescribe

What it does

mDescribe returns all available methods within the WinInfo Xobject

Parameters

none

Lingo Examples

```
openXLib "WININFO.DLL"
put wininfo(mDescribe)
closeXLib "WININFO.DLL"
```

What it returns

mDescribe returns all available methods within the WinInfo Xobject:

```
-- WinInfo XObject v1.0 Jan 1 1995 (c)1995 createMedia
I     mNew               -- Creates a new instance of the XObject
X     mDispose           -- Disposes of XObject instance
S     mName              -- Returns the XObject name (WinInfo)
S     mWindowsDirectory  -- Returns windows directory path
S     mSystemDirectory   -- Returns system directory path
SI    mGetDriveType, DriveNumber -- A=0,B=1,etc. Returns type
SSSSS mReadPrivateProfile, Section, Entry, Default, FileName
ISSSS mWritePrivateProfile, Section, Entry, NewValue, FileName
S     mWindowsVersion    -- Returns Windows version
S     mDOSVersion        -- Returns DOS version
I     mScreenWidth       -- Returns width of screen
I     mScreenHeight      -- Returns height of screen
```

mWindowsDirectory

What it does

returns the Windows directory (where WIN.COM is located)

Parameters

none

Lingo Examples

```
global gWinDir
openXLib "WININFO.DLL"
set obj = wininfo(mNew)
set gWinDir = obj(mWindowsDirectory)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

returns the Windows directory, for example:
"C:\WINDOWS"

mSystemDirectory

What it does

returns the Windows System directory (where USER.EXE et. al. system files are located)

Parameters

none

Lingo Examples

```
global gSysDir
openXLib "WININFO.DLL"
set obj = wininfo(mNew)
set gSysDir = obj(mSystemDirectory)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

returns the system directory, for example:
"C:\WINDOWS\SYSTEM"

mWindowsVersion

What it does

returns the Windows version.

Parameters

none

Lingo Examples

```
global gWinVer
openXLib "WININFO.DLL"
set obj = wininfo(mNew)
set gWinVer = obj(mWindowsVersion)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

returns the Windows version, for example:
"Windows version 3.10"

mDOSVersion

What it does

returns the DOS version.

Parameters

none

Lingo Examples

```
global gDOSVer
openXLib "WININFO.DLL"
set obj = wininfo(mNew)
set gDOSVer = obj(mDOSVersion)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

returns the DOS version, for example:
"MS-DOS version 6.1"

mGetDriveType

What it does

returns the type (Floppy, Fixed, Network, etc.) of the selected drive

Parameters

driveNumber as integer, where A:= 0, B:= 1, C:= 2, D:= 3, etc.

Lingo Examples

```
openXLib "WININFO.DLL"
set driveNumber = 3 -- check drive type of drive D
set obj = wininfo(mNew)
put obj(mGetDriveType, driveNumber)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

"Floppy" if floppy drive
"Fixed" if fixed or hard drive
"Network" if network drive
"CD-ROM" if CD-ROM drive
"RAM Disk" if RAM Disk
"Removable" if SyQuest or some other type of removable drive
"Undetermined" if unable to determine drive type

mScreenWidth

What it does

returns the width of the current screen, independent of any movie stage size

Parameters

none

Lingo Examples

```
openXLib "WININFO.DLL"
set obj = wininfo(mNew)
put obj(mScreenWidth)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

640 if screen width is 640 pixels;
800 if screen width is 800 pixels; etc.

mScreenHeight

What it does

returns the height of the current screen, independent of any movie stage size

Parameters

none

Lingo Examples

```
openXLib "WININFO.DLL"
set obj = wininfo(mNew)
put obj(mScreenHeight)
obj(mDispose)
closeXLib "WININFO.DLL"
```

What it returns

480 if screen height is 480 pixels;
600 if screen height is 600 pixels; etc.

mReadPrivateProfile

What it does

reads an entry value from a private profile (initialization or INI) file.

Parameters

Section string identifying INI file section
Entry string identifying INI file entry
Default string identifying default entry value if none exists
Filename string identifying path to INI file

Lingo Examples

```
        -- read a preference setting from MYAPP.INI
        --
        openXLib "WININFO.DLL"
        set obj = wininfo(mNew)
        set lSection = "SETTINGS"
        set lEntry = "USERLEVEL"
        set lDefault = "BEGINNER"
        set lFileName = "MYAPP.INI"
        set Setting = obj(mReadPrivateProfile, lSection,lEntry, lDefault,
lFileName)
        obj(mDispose)
        closeXLib "WININFO.DLL"
```

What it returns

String of provided entry value or the defailt parameter if no entry in INI file.

Notes

This method searches the file specified in the Filename paramater for an entry that matches the name specified by the Entry parameter under the section heading specified by the Section parameter. If the entry is found, its corresponding string is returned. If the entry does not exist, the default character string specified by the Default parameter assigned to the entry and the default value is returned.

If no path to the filename is provided, WinInfo looks for the file in the Windows directory.

A string entry in the initialization file must have the following form:

[section]
entry=string
.
.
.

mWritePrivateProfile

What it does

writes an entry value to a private profile (initialization or INI) file.

Parameters

Section string identifying INI file section
Entry string identifying INI file entry

NewValue string identifying new entry value
Filename string identifying path to INI file

Lingo Examples

```
      -- writes a preference setting to MYAPP.INI
      openXLib "WININFO.DLL"
      set obj = wininfo(mNew)
      set lSection = "SETTINGS"
      set lEntry = "USERLEVEL"
      set lNewValue = "ADVANCED"
      set lFileName = "MYAPP.INI"
      set iReturn = obj(mWritePrivateProfile, lSection,lEntry,
lNewValue, lFileName)
      obj(mDispose)
      closeXLib "WININFO.DLL"
```

What it returns

1 if succesful;
0 if unsuccesful

Notes

This method searches the file specified in the Filename paramater for the section heading specified by the Section parameter. If the section is found, its corresponding entry specified bu the Entry parameter is updated with the value specified in the NewValue paramater.

If the file is not found a new file is created. If no path to the filename is provided, WinInfo creates the file in the Windows directory.

If the section is not found, a new section is created.

If the entry is not found, a new entry is created.

If no path to the filename is provided, WinInfo looks for the file in the Windows directory.

A string entry in the initialization file must have the following form:

```
[section]
entry=string
    .
    .
    .
```

Licensing Information

In exchange for the licensing fee of $95.00 (US funds), the licensee is granted permission to distribute the WinInfo XObject as a component of one commericial software product. WinInfo may be distributed in unlimited quantities with the product and future upgrades to the same product. The licensed version of WinInfo does not contain the splash screen displayed when creating a WinInfo object.

Non-commericial use of WinInfo for personal purposes is free.

To license WinInfo, contact:

createMedia
25 Barry Lane
Simsbury, CT 06070 USA

203-651-3345 (voice)
203-651-3345 (fax)

74444,1574@compuserve.com
createMed@aol.com

WinInfo is provided "as is." This license agreement is not a guarantee of WinInfo's suitability for any particular purpose, nor a guarantee of compatibility with present or future versions of system software, system extensions, etc. The licensee assumes all responsibility for testing and quality assurance of any products using the WinInfo XObject.

Additional Products in Development

Currently under development is an XObject to manage Windows' ScreenSaver and Task Switching: on/off status and the ability to enable/disable. Any thoughts or comments? Let us know!