

# Registry.vcx

Visual FoxPro 6.0 contiene las clases nuevas de Foundation (FFC) que ofrecen compatibilidad para una amplia variedad de funciones de base de datos y aplicación. Una de las tareas más importantes de las aplicaciones de hoy es la posibilidad de tener acceso al registro. El FFC incluye una biblioteca de clase denominada Registry.vcx, que contiene clases útiles para obtener acceso al registro. Este documento describe formas de utilizar estas clases en las aplicaciones.

Puede utilizar la biblioteca de clase Registry.vcx incluido en la carpeta \FFC de Visual FoxPro 6.0 para tener acceso al registro de Windows. Las clases incluidas en Registry.vcx, acceso al registro de llamadas utilizando el comando DECLARE de Visual FoxPro para llamar directamente a las rutinas de API Win32. La biblioteca de clase consta de las siguientes clases:

## Registro

Ésta es la clase principal que contiene el registro de la mayoría de los métodos que necesite tener acceso (lectura y escritura). Hay muchos fines útiles para el acceso al registro en una aplicación.

En el ejemplo siguiente se recorre el registro para obtener una lista de controles ActiveX registrados instalados en el sistema. Los resultados se almacenan en una matriz pública denominada aControls:

```
cOptPath = "CLSID"
public aControls
DIMENSION aControls[1,2]
STORE "" TO aControls
DIMENSION aCLSIDs[1]

oReg = NewObject("Registry",HOME()+"FFC\REGISTRY")
IF oReg.EnumOptions(@aCLSIDs,cOptPath,"",.T.) # 0
    RETURN .F.
ENDIF
FOR i = 1 TO ALEN(aCLSIDs)
    IF m.i%10 = 0
        WAIT WINDOW NOWAIT "Processed "+STR(m.i)+" of "+STR(ALEN(aCLSIDs))
    ENDIF
    DIME aKeys[1]
    DIME aProgID[1,2]
    DIME aControlName[1,2]
    STORE "" TO aProgID,aControlName

    IF oReg.EnumOptions(@aKeys,cOptPath+"\ "+aCLSIDs[m.i],"",.T.) = 0
        nPos = ASCAN(aKeys,"Control")
        IF m.nPos = 0
            LOOP
        ENDIF
        IF !EMPTY(aControls[1,2])
            DIMENSION aControls[ALEN(aControls,1)+1,2]
        ENDIF
        oReg.EnumOptions(@aControlName,cOptPath+"\ "+aCLSIDs[m.i])
        oReg.EnumOptions(@aProgID,cOptPath+"\ "+aCLSIDs[m.i)+"\ProgID")
        aControls[ALEN(aControls,1),1] = aControlName[2]
        aControls[ALEN(aControls,1),2] = aProgID[2]
    ENDIF
ENDFOR
WAIT CLEAR
DISPLAY MEMORY LIKE aControls

* To view all the controls, open the debugger and enter aControls in the
* Watch window.
```

## FileReg

Esto es una subclase del registro que tiene varios métodos útiles para tener acceso a información de archivo específica como ruta de acceso y la versión. En el ejemplo siguiente se muestra cómo obtener acceso a información de la ruta de acceso y la versión para Excel:

```
oReg = NewObject("FileReg",HOME()+"FFC\REGISTRY")
STORE "" TO cExtKey,cAppKey
IF oReg.GetAppPath("XLS",@cExtKey,@cAppKey) = 0
    ? cExtKey    && ex. Excel.Sheet.8
    ? cAppKey    && ex. "C:\Office\excel.exe" /e
ENDIF
```

## Foxreg

Esto es una subclase del registro y tiene métodos que la lectura y escritura a las claves del registro de opciones de Visual FoxPro. Estos valores se guardan desde el diálogo de opciones disponible fuera del menú Herramientas. Aunque la mayoría de estas opciones sólo se asigna a comandos SET, unos cuantos sólo se almacenan en el registro (área de diseño de formulario máximo). El fragmento de código siguiente devuelve el valor actual de BuilderLock (bloqueado 1 & 0 no está bloqueado):

```
oReg = NewObject("FoxReg",HOME()+"FFC\REGISTRY")
cOptionValue = ""
cOptionName = "BuilderLock"
IF oReg.GetFoxOption(m.cOptionName,@cOptionValue)=0
    ? cOptionValue
ENDIF
```

## Odbcreg

Esta subclase del registro tiene funciones de acceso a claves ODBC para las aplicaciones que requieren dicha información. El código siguiente enumera en la pantalla todos los controladores ODBC y orígenes de datos en el sistema.

```
oReg = NewObject("ODBCReg",HOME()+"FFC\REGISTRY")
DIMENSION aODBCDrivers[1]
DIMENSION aODBCData[1]
nErrNum = oReg.GetODBCDrvr(@aODBCDrivers)
DISPLAY MEMO LIKE aODBCDrivers
nErrNum = oReg.GetODBCDrvr(@aODBCData,.T.)
DISPLAY MEMO LIKE aODBCData
```

## Oldinireg

Esta subclase del registro se incluye por compatibilidad para tener acceso a los archivos de texto que tienen el formato antiguo de INI.

Suponga que tiene una copia de un archivo Win.ini antiguo en el directorio Windows.

En el ejemplo siguiente se devuelve el valor de la entrada de dispositivo en la sección windows de un archivo Win.ini:

```
LOCAL rcValue
x=NEWOBJECT("oldinireg",HOME()+"FFC\REGISTRY")
x.GetIniEntry(@rcValue,"windows","device","c:\windows\win.ini")
IF !empty(rcValue)
    ?rcValue
ELSE
    ?"Value not found"
ENDIF
```

el ejemplo siguiente escribe un valor en la entrada de ejecución de la sección de Windows del archivo Win.ini:

```
local rcValue
rcValue = "test.exe"
x=NEWOBJECT("oldinireg",HOME()+"FFC\REGISTRY")
x.WriteIniEntry(rcValue,"windows","run","d:\windows\win.ini")
x.GetIniEntry(@rcValue,"windows","run","d:\windows\win.ini")
?rcValue
```