ApiSimple – An ActiveX control for COBOL Sp2 Programmers

What is ApiSimple?

ApiSimple is an ActiveX control developed specifically for Sp2 programmers that is designed to encapsulate certain Windows API calls that COBOL Sp2 Programmers find useful. In addition to exposing specific API calls, ApiSimple also provides some additional functionality at a higher level.

For example, to delete all files in a particular directory takes a complex set of API calls, but ApiSimple handles this for you – allowing you to simply pass the directory you want to have all files (or a specified set of files based on wild cards) deleted from.

The feature set of ApiSimple is fairly diverse, but is limited to non user interface features. ApiSimple does not add new types of fields, icons, objects, widgets or controls to your application – it works under the covers.

Why is something like ApiSimple needed?

The windows API is vast and complex. However, quite frequently a task required for your program is most easily accomplished by making API calls. These calls can make things like starting an external program easier and more consistent. For example – the single method of using the ShellExecute API works to start programs on all versions of windows, while using Vendor supplied external shells have different requirements for different versions of windows.

The learning curve for using the Windows API is fairly steep. Just figuring out what is available is time consuming. ApiSimple includes many of the more commonly requested API functions.

Why not just use API calls directly from the COBOL program?

There are a couple of reasons to avoid API calls directly form your COBOL program:

1. API calls are complex and not documented for COBOL programmers. Just getting them to work can be very time consuming
2. If you use thin client, you might need to execute these API calls on the client side. This is impossible if you make the API calls directly from your COBOL program.

3. Each COBOL compiler implements calls to the API (when implemented at all) in a different manner. If you make direct API calls in your COBOL program you are locked into a specific compiler.
4. When used with thin client, ApiSimple lets you execute WindowsAPI calls on the client. The server could even be a Unix or VMS server!

Is any knowledge of the Windows API required to use ApiSimple?

No specific API knowledge is necessary to use ApiSimple. All you need to know is what you want to do. Examples in this document show at least one use for each of the supported API functions.

What are the license restrictions for ApiSimple?

Once licensed you are free to distribute the ApiSimp.ocx file with your applications. Only users current on their Sp2 Maintenance and have a license to use the Sp2 ActiveX add on (sp2ocx32.dll) may license the control.

Does the control have any other software dependencies?

Not for the control. The control is fairly large in size because all of the software dependencies are linked into the control. The control works on all 32 bit versions of Windows with no additional software or required libraries.

However, to use the control with Sp2 you will have to have licensed the Sp2 ActiveX add on dll – Sp2ocx32.dll.

What happens if there is an update to the ApiSimple Control?

Once licensed, and if your maintenance is kept current, you will be able to obtain updates as they are released.

What happens if there is a function that I think might be right for ApiSimple, but it doesn't support it?

Email us your request. We are always willing to assist COBOL programmers in accomplishing their desired goals with ease. We will enhance ApiSimple as necessary to help you accomplish your goals.

New Methods/Properties

Versoin 1.0.1.5

No new functionality. Fix findFile problem where first filename was not returned.

Version 1.0.1.4

apiCreateDirectory – Creates the specified directory

Version 1.0.1.3

findFile – returns a count of file names matching the specified mask and folder.
retFileName – returns a file name retrieved via findFile, when passed the occurance of that file

Version 1.0.1.1

apiSnapshot – Capture current screen or window to the clipboard
showOption – Control the way the window is displayed for processes started via the Shell or ShellWait methods.
apiGetCurrentDirectory – returns the current directory

What are the components of ApiSimple and how do I use them?

ApiSimple consists of the control itself: ApiSimp.OCX. A copybook and panel are provided for the ApiSimple functions. Sample programs are provided illustrating some of the various functions of the control.

Getting Started

Use the ApiSimple ActiveX control like any other ActiveX control used with sp2. If you add the control to a panel it should register automatically. You can also register the control by using regsvr32 apisimp.ocx or by using the ApiSimp.exe which is a setup file that will install the control in the folder from which it is run. The setup may be run silently with no user intervention by specifying "ApiSimp.exe /s" on the command line.

You do not need to install the control in the windows system folder, although you may if you desire.

When you add the control to a panel, you will see a circle drawn. This is performed only so you can see that the control is valid, registered and operating properly. You may hide the control or size it with a height and width of zero. It is suggested that you position the control on row zero column zero of any panel you place it on.

What functions are available with ApiSimple

The basic functions available are described below. Some of these functions can be used for multiple purposes. For example, the Mutex functions can be used to ensure that only a single copy of an application is running and if another is running you can use the apiShowWindow method to show the window from the other copy of the running application before shutting down.

1. Basic Async communications
2. Mutex creation and detection
3. Find a window by title (returns handle)
4. Show a window by handle
5. Grab focus (by window handle)
6. Get the logged on users user name
7. Get the computers name
8. Get the network card address for all network cards installed
9. Compute an MD5 hash from a string
10. Read and write binary files
11. Get information about a file
    a. Find a file
    b. Get date/time/size
12. Update a files last update date
13. Delete a file
14. Delete files using wild cards
15. Determine the current screen resolution
16. Copy the image on the clipboard to a BMP file
17. Copy the image on the clipboard to a JPG file
18. Convert a BMP to a JPG file
19. Get version information about an exe, dll or ocx file. (Same thing that shows when you right click and choose Properties from Explorer)
20. Kill a process when you know the window title of one of the windows associated with the process
21. Access a shared memory block from different running programs
22. Shell – Execute a program
    a. Actually more powerful that just program execution.
        i. If you pass the name of a PDF file it will open the PDF with Acrobat.
        ii. If you pass a URL for a web site, it opens the default web browser

        iii. You can pass mailto:someone@somewhere.com and the default mail client will be started with the To: address filled in

    b. A shell function is available that either waits for the started process to complete or returns immediately

23. Find A file
24. Determine Current Directory
25. Create a Directory

Methods, Properties and Examples

Lets begin by reviewing the differences between methods and properties. With ActiveX controls properties are data values that can be retrieved. Methods are procedures that are called. Typically methods have one or more parameters, but not necessarily so. Properties generally have no parameters and can be used to set or get a property value. With methods, even when method names indicate that they “get” something, you will use an Sp2-Set-OCX call to invoke the methods.

Generally speaking, ApiSimple will return status values from those methods that return a status as “Success” or “Fail”. Additional text may appear after the word “Success” or “Fail” detailing other aspects of the result of the method invocation.

Methods and Properties available with ApiSimple

The methods that follow are organized in a logical sequence since some methods are used in isolation and some are used in conjunction with others.

NOTE: The SP2-OCX-DEFINITIONS shown here assume an unmodified sp2ocx.cpy file. Many of the functions of ApiSimple – particularly the file handling examples – can benefit from using a modified sp2ocx.cpy. Included with ApiSimple is a USROCX.CPY which has expanded sizes for the various values. When using the expanded sizes, things like *methodname-parmdesc* (The data area for the parameters passed) might not be long enough in the SP2-OCX-DEFINITION shown here.

In addition, some of the data names generated for the definitions may be too long for some compilers to handle. You can either shorten these names when you encounter an error, or use the included api-simp-short.cpy instead of api-simp.cpy.

apiGetComputerName (Method)

This method returns the name associated with the computer. When computers are part of a network, their names must be unique. However, since your users may or may not be part of a network you cannot count on this being a completely unique name.

Sp2-OCX-Definition:

```
01  api-apiGetComputerName.
    05  apiGetComputerName-prop-name
                    pic x(30) value "apiGetComputerName".
    05  apiGetComputerName-access-type
                    pic s9(4) comp-5 value +0001.
    05  apiGetComputerName-parm-cnt
                    pic s9(4) comp-5 value +0000.
    05  apiGetComputerName-value-type.
        10  xString  pic s9(4) comp-5 value +0008.
        10  xEmpty   pic s9(4) comp-5 value +0000.
        10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiGetComputerName-Prop-Name to Sp2-OC-Prop-Name
Move apiGetComputerName-Access-Type to Sp2-OC-Access-Type
Move apiGetComputerName-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiGetComputerName-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

apiGetUserName (Method)

This method returns the logged on user name. On some Windows computers, windows returns a “garbage” name if a network logon is not performed.

Sp2-OCX-Definition:

```
01  api-apiGetUserName.
    05  apiGetUserName-prop-name
```

```
                pic x(30) value "apiGetUserName".
      05  apiGetUserName-access-type
                pic s9(4) comp-5 value +0001.
      05  apiGetUserName-parm-cnt
                pic s9(4) comp-5 value +0000.
      05  apiGetUserName-value-type.
         10  xString  pic s9(4) comp-5 value +0008.
         10  xEmpty   pic s9(4) comp-5 value +0000.
         10  filler   pic s9(4) comp-5 value +0000.
         10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiGetUserName-Prop-Name to Sp2-OC-Prop-Name
Move apiGetUserName-Access-Type to Sp2-OC-Access-Type
Move apiGetUserName-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiGetUserName-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

apiGetMac (Method)

This method retrieves the MAC addresses that are known. The MAC address is a unique global identifier associated with an Ethernet network adapter. Dialup networking assigns a virtual MAC address as well that may not be unique. The method execution returns a number of MAC addresses found. When multiple networking protocols are associated with a particular MAC address, the same address may be returned more than once. After using this method to retrieve the number of Mac Addresses, you must use apiRetMac to actually find out the values of the addresses returned.

Note: Requires windows based networking to be installed. If IP is the only protocol installed and no sharing is specified, the MAC address may not be returned.

Sp2-OCX-Definition:

```
    01  api-apiGetMac.
      05  apiGetMac-prop-name
```

```
                pic x(30) value "apiGetMac".
      05  apiGetMac-access-type
                pic s9(4) comp-5 value +0001.
      05  apiGetMac-parm-cnt
                pic s9(4) comp-5 value +0000.
      05  apiGetMac-value-type.
         10  xInteger pic s9(4) comp-5 value +0002.
         10  xEmpty   pic s9(4) comp-5 value +0000.
         10  filler   pic s9(4) comp-5 value +0000.
         10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiGetMac-Prop-Name to Sp2-OC-Prop-Name
Move apiGetMac-Access-Type to Sp2-OC-Access-Type
Move apiGetMac-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiGetMac-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

Since SP2-OC-VALUE is an alphanumeric field, you will need to convert the returned value to a number.

Ie: Compute NUM-RETURNED = Function NUMVAL (Sp2-OC-Value)

apiRetMac (Method)

This method returns the specified MAC address which was retrieved via a call to apiGetMac. It accepts a single parameter which is the number of the MAC address to return. When apiGetMac is called, it returns the number of MAC addresses found. If 3 MAC addresses are found, apiRetMac can be called 3 times – with 1, 2 and 3 respectively as the parameter in order to retrieve the 3 MAC addresses found.

Sp2-OCX-Definition:

```
    01  api-apiRetMac.
      05  apiRetMac-prop-name
                pic x(30) value "apiRetMac".
```

```
         05  apiRetMac-access-type
                      pic s9(4) comp-5 value +0001.
         05  apiRetMac-parm-cnt
                      pic s9(4) comp-5 value +0001.
         05  apiRetMac-value-type.
             10  xString  pic s9(4) comp-5 value +0008.
             10  xEmpty   pic s9(4) comp-5 value +0000.
             10  filler   pic s9(4) comp-5 value +0000.
             10  filler   pic x(4) value low-values.
         05  apiRetMac-parms.
             10  apiRetMac-item pic x(80).
             10  apiRetMac-item-type.
                 15  xInteger pic s9(4) comp-5 value +0002.
                 15  xEmpty   pic s9(4) comp-5 value +0000.
                 15  filler   pic s9(4) comp-5 value +0000.
                 15  filler   pic x(4) value low-values.
```

Example:

```
01  Number-To-Retrieve Pic 9(3).

Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiRetMac-Prop-Name to Sp2-OC-Prop-Name
Move apiRetMac-Access-Type to Sp2-OC-Access-Type
Move apiRetMac-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiRetMac-Value-Type to Sp2-OC-Value-Type
Move number-to-retrieve To ApiRetMac-Item
Move apiRetMac-Item to Sp2-OC-Parm-Value (1)
Move apiRetMac-Item-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

If you try to retrieve an item number that doesn't exist "Error" will be returned in SP2-OC-VALUE.

Note: Since parm values are always passed as alphanumeric data items, ensure that when passing numbers, such as the Number-To-Retrieve that the COBOL definition is that of a numeric edited item or usage display unsigned number, and not a numeric item that is packed, binary or defined with an embedded sign.

apiGetScreenRes (Method)

This method retrieves the screen resolution. ActiveX controls can only return a single value, so this method places the horizontal and vertical resolution into properties that can be retrieved after invoking the method.

Sp2-OCX-Definition:

```
    01  api-apiGetScreenRes.
        05  apiGetScreenRes-prop-name
                      pic x(30) value "apiGetScreenRes".
        05  apiGetScreenRes-access-type
                      pic s9(4) comp-5 value +0001.
        05  apiGetScreenRes-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  apiGetScreenRes-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiGetScreenRes-Prop-Name to Sp2-OC-Prop-Name
Move apiGetScreenRes-Access-Type to Sp2-OC-Access-Type
Move apiGetScreenRes-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiGetScreenRes-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The result returned will be “Success” or “Fail”. If the resolution was successfully retrieved the status returned will be “Success”.

After invoking the method, the properties hRes and vRes contain the horizontal and vertical resolution.

hRes (Property)

This property will contain the horizontal resolution in DPI (Dots Per Inch) after a successful call to apiGetScreenRes.

Sp2-OCX-Definition:

```
      01  api-hRes.
          05  hRes-prop-name
                          pic x(30) value "hRes".
          05  hRes-access-type
                          pic s9(4) comp-5 value +0000.
          05  hRes-parm-cnt
                          pic s9(4) comp-5 value +0000.
          05  hRes-value-type.
              10  xInteger pic s9(4) comp-5 value +0002.
              10  xEmpty   pic s9(4) comp-5 value +0000.
              10  filler   pic s9(4) comp-5 value +0000.
              10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move hRes-Prop-Name to Sp2-OC-Prop-Name
Move hRes-Access-Type to Sp2-OC-Access-Type
Move hRes-Parm-Cnt to Sp2-OC-Parm-Cnt
Move hRes-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

Since SP2-OC-VALUE is an alphanumeric field, you will need to convert the returned value to a number.

Ie: Compute Horiz-Resolution = Function NUMVAL (Sp2-OC-Value)

vRes (Property)

This property will contain the vertical resolution in DPI (Dots Per Inch) after a successful call to apiGetScreenRes.

Sp2-OCX-Definition:

```
      01  api-vRes.
          05  vRes-prop-name
                          pic x(30) value "vRes".
          05  vRes-access-type
                          pic s9(4) comp-5 value +0000.
```

```
    05  vRes-parm-cnt
                 pic s9(4) comp-5 value +0000.
    05  vRes-value-type.
        10  xInteger pic s9(4) comp-5 value +0002.
        10  xEmpty   pic s9(4) comp-5 value +0000.
        10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move vRes-Prop-Name to Sp2-OC-Prop-Name
Move vRes-Access-Type to Sp2-OC-Access-Type
Move vRes-Parm-Cnt to Sp2-OC-Parm-Cnt
Move vRes-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

Since SP2-OC-VALUE is an alphanumeric field, you will need to convert the returned value to a number.

Ie: Compute Vert-Resolution = Function NUMVAL (Sp2-OC-Value)

apiSnapshot (Method) – Available as of version 1.0.1.1

This method captures either the entire screen or the window that currently has focus to the Windows clipboard. The clipToBmp and clipToJpg methods can then be used to save the image to a file.

Sp2-OCX-Definition:

```
 01  api-apiSnapShot.
    05  apiSnapShot-prop-name
                 pic x(80) value
       "apiSnapShot".
    05  apiSnapShot-access-type
                 pic s9(4) comp-5 value +0001.
    05  apiSnapShot-parm-cnt
                 pic s9(4) comp-5 value +0001.
    05  apiSnapShot-value-type.
       10  xVoid       pic s9(4) comp-5 value +0024.
       10  xEmpty      pic s9(4) comp-5 value +0000.
```

```
            10  filler       pic s9(4) comp-5 value +0000.
            10  filler       pic x(4) value low-values.
         05  apiSnapShot-parms.
     * Name Shortened - was:
     * apiSnapShot-ScreenOrWindow
            10  apiSnapShot-ScreenOrWindw pic x(80).
            10  apiSnapShot-ScreenOrWindw-type.
               15  xByte      pic s9(4) comp-5 value +0017.
               15  xEmpty     pic s9(4) comp-5 value +0000.
               15  filler     pic s9(4) comp-5 value +0000.
               15  filler     pic x(4) value low-values.
```

The parm (ScreenOrWindow) should contain 0 to capture the entire screen or a 1 to capture just the current window.

Example:

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiSnapshot-Prop-Name to Sp2-OC-Prop-Name
Move apiSnapshot-Access-Type to Sp2-OC-Access-Type
Move apiSnapshot-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiSnapshot-Value-Type to Sp2-OC-Value-Type
Move “0” To apiSnapshot-ScreenOrWindw
Move apiSnapshot-ScreenOrWindw to Sp2-OC-Parm-Value (1)
Move apiSnapShot-ScreenOrWindw-type to Sp2-OC-Parm-Type (1)
Call “SP2” using sp2-set-ocx sp2-ocx-parm
Call “SP2” using sp2-display-window sp2-null-parm
```

Nothing is returned.

****IMPORTANT NOTE! – In order for the image capture to release a lock on the clipboard, you must issue a display-window call immediately after the Set-OCX call as shown in this example.

bmpToJpg (Method)

This method converts a BMP format image file to a JPG image file. This can be used in conjunction with the Flexus JPG add on for Sp2 and Form Print so as to reduce the size of image files. JPG files are typically much smaller than BMP’s.

Sp2-OCX-Definition:

```
01  api-bmpToJpg.
    05  bmpToJpg-prop-name
                  pic x(30) value "bmpToJpg".
    05  bmpToJpg-access-type
                  pic s9(4) comp-5 value +0001.
    05  bmpToJpg-parm-cnt
                  pic s9(4) comp-5 value +0002.
    05  bmpToJpg-value-type.
        10  xString  pic s9(4) comp-5 value +0008.
        10  xEmpty   pic s9(4) comp-5 value +0000.
        10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
    05  bmpToJpg-parms.
        10  bmpToJpg-bmpFileName pic x(80).
        10  bmpToJpg-bmpFileName-type.
            15  xString  pic s9(4) comp-5 value +0008.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
        10  bmpToJpg-jpgFileName pic x(80).
        10  bmpToJpg-jpgFileName-type.
            15  xString  pic s9(4) comp-5 value +0008.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
```

The method accepts two parameters. The first is the fully qualified BMP file name that is to be converted. The second is the fully qualified JPG file that will result. If the JPG already exists, it will be overwritten.

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move bmpToJpg-Prop-Name to Sp2-OC-Prop-Name
Move bmpToJpg-Access-Type to Sp2-OC-Access-Type
Move bmpToJpg-Parm-Cnt to Sp2-OC-Parm-Cnt
Move bmpToJpg-Value-Type to Sp2-OC-Value-Type
Move “C:\bmpfile.bmp” TO bmpToJpg-bmpFileName
Move “C:\bmpfile.jpg” TO bmpToJpg-jpgFileName
Move bmpToJpg-bmpFileName to Sp2-OC-Parm-Value (1)
Move bmpToJpg-bmpFileName-Type to Sp2-OC-Parm-Type (1)
Move bmpToJpg-jpgFileName to Sp2-OC-Parm-Value (2)
Move bmpToJpg-jpgFileName-Type to Sp2-OC-Parm-Type (2)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

The returned value in SP2-OC-VALUE will be "Success" if the image file was converted, or "Fail" if the conversion failed – normally caused by an invalid file name in either the source or the destination.

clipToBmp (Method)

This method takes an image that is on the Windows clipboard and stores it as a Bmp. There are several ways to get the image on the clipboard. One easy method is to hit the print screen key on the keyboard. ALT+Print Screen copies just the current window. Some image software packages can copy images to the clipboard as well.

SP2-OCX-DEFINITION:

```
    01  api-clipToBmp.
        05  clipToBmp-prop-name
                      pic x(30) value "clipToBmp".
        05  clipToBmp-access-type
                      pic s9(4) comp-5 value +0001.
        05  clipToBmp-parm-cnt
                      pic s9(4) comp-5 value +0001.
        05  clipToBmp-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  clipToBmp-parms.
            10  clipToBmp-fileName pic x(80).
            10  clipToBmp-fileName-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

Example:

The method accepts a single parameter – the name of the image file to store. Regardless of the file name you specify, the image will be saved in BMP format.

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move clipToBmp-Prop-Name to Sp2-OC-Prop-Name
Move clipToBmp-Access-Type to Sp2-OC-Access-Type
Move clipToBmp-Parm-Cnt to Sp2-OC-Parm-Cnt
Move clipToBmp-Value-Type to Sp2-OC-Value-Type
Move “C:\bmpfile.bmp” TO ClipToBmp-FileName
Move clipToBmp-FileName to Sp2-OC-Parm-Value (1)
Move clipToBmp-FileName-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The returned value in SP2-OC-VALUE will be “Success” if the image file was created, or “Fail” if the creation failed – normally caused by an invalid file name in either the source or the destination.

clipToJpg (Method)

This method takes an image that is on the Windows clipboard and stores it as a Jpg. There are several ways to get the image on the clipboard. One easy method is to hit the print screen key on the keyboard. ALT+Print Screen copies just the current window. Some image software packages can copy images to the clipboard as well.

SP2-OCX-DEFINITION:

```
    01  api-clipToJpg.
        05  clipToJpg-prop-name
                        pic x(30) value "clipToJpg".
        05  clipToJpg-access-type
                        pic s9(4) comp-5 value +0001.
        05  clipToJpg-parm-cnt
                        pic s9(4) comp-5 value +0001.
        05  clipToJpg-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  clipToJpg-parms.
            10  clipToJpg-fileName pic x(80).
            10  clipToJpg-fileName-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
```

```
               15  filler   pic s9(4) comp-5 value +0000.
               15  filler   pic x(4) value low-values.
```

Example:

The method accepts a single parameter – the name of the image file to store. Regardless of the file name you specify, the image will be saved in JPG format.

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move clipToJpg-Prop-Name to Sp2-OC-Prop-Name
Move clipToJpg-Access-Type to Sp2-OC-Access-Type
Move clipToJpg-Parm-Cnt to Sp2-OC-Parm-Cnt
Move clipToJpg-Value-Type to Sp2-OC-Value-Type
Move “C:\bmpfile. jpg” TO ClipToJpg-FileName
Move clipToJpg-FileName to Sp2-OC-Parm-Value (1)
Move clipToJpg-FileName-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The returned value in SP2-OC-VALUE will be “Success” if the image file was created, or “Fail” if the creation failed – normally caused by an invalid file name in either the source or the destination.

apiCreateMutex (Method)

This method creates a Windows Mutex named the same as the passed name. A “Mutex” is like a semaphore. When created, the named mutex exists until the application that creates it goes away. A mutex can be used to check to see if another copy of your process is already running so that you can have your program behave appropriately in that case. See the CHECKACT.CBL example for an example of apiCreateMutex and it’s use in a real world situation.

SP2-OCX-DEFINITION:

```
    01  api-apiCreateMutex.
        05  apiCreateMutex-prop-name
                      pic x(30) value "apiCreateMutex".
        05  apiCreateMutex-access-type
                      pic s9(4) comp-5 value +0001.
        05  apiCreateMutex-parm-cnt
                      pic s9(4) comp-5 value +0001.
        05  apiCreateMutex-value-type.
```

```
          10  xLong    pic s9(4) comp-5 value +0003.
          10  xEmpty   pic s9(4) comp-5 value +0000.
          10  filler   pic s9(4) comp-5 value +0000.
          10  filler   pic x(4) value low-values.
      05  apiCreateMutex-parms.
          10  apiCreateMutex-MutexName pic x(80).
          10  apiCreateMutex-MutexName-type.
              15  xString  pic s9(4) comp-5 value +0008.
              15  xEmpty   pic s9(4) comp-5 value +0000.
              15  filler   pic s9(4) comp-5 value +0000.
              15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiCreateMutex-Prop-Name to Sp2-OC-Prop-Name
Move apiCreateMutex-Access-Type to Sp2-OC-Access-Type
Move apiCreateMutex-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiCreateMutex-Value-Type to Sp2-OC-Value-Type
Move “MyMutexName” TO apiCreateMutex-MutexName
Move apiCreateMutex-MutexName to Sp2-OC-Parm-Value (1)
Move apiCreateMutex-MutexName-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Unlike other functions in ApiSimple, this function returns a numeric value in SP2-OC-VALUE. Like other numeric values, a NUMVAL function can be used to convert it from text to a number.

Values:

0 = The Mutex did not exist and was created
183= The Mutex already exists

In addition, the Windows handle to the Mutex is saved in the apiSimple Property:

apiDataLastHandle (Property)

The handle can be retrieved using a sp2-oc-get call against this property.

SP2-OCX-DEFINITION:

```
    01  api-apiDataLastHandle.
      05  apiDataLastHandle-prop-name
                  pic x(30) value "apiDataLastHandle".
      05  apiDataLastHandle-access-type
                  pic s9(4) comp-5 value +0000.
      05  apiDataLastHandle-parm-cnt
                  pic s9(4) comp-5 value +0000.
      05  apiDataLastHandle-value-type.
         10  xLong    pic s9(4) comp-5 value +0003.
         10  xEmpty   pic s9(4) comp-5 value +0000.
         10  filler   pic s9(4) comp-5 value +0000.
         10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiDataLastHandle-Prop-Name to Sp2-OC-Prop-Name
Move apiDataLastHandle-Access-Type to Sp2-OC-Access-Type
Move apiDataLastHandle-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiDataLastHandle -Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

apiFindWindow (Method)

This method finds a window when passed the window title and returns it’s window handle. This window handle can be used with the apiShowWindow function to show a window that might be minimized or otherwise hidden on the desktop. An example of it’s use in found in the CHECKACT.CBL sample which is included.

SP2-OCX-DEFINITION:

```
    01  api-apiFindWindow.
      05  apiFindWindow-prop-name
                  pic x(30) value "apiFindWindow".
      05  apiFindWindow-access-type
                  pic s9(4) comp-5 value +0001.
      05  apiFindWindow-parm-cnt
                  pic s9(4) comp-5 value +0001.
      05  apiFindWindow-value-type.
      10  xLong    pic s9(4) comp-5 value +0003.
      10  xEmpty   pic s9(4) comp-5 value +0000.
      10  filler   pic s9(4) comp-5 value +0000.
      10  filler   pic x(4) value low-values.
   05  apiFindWindow-parms.
      10  apiFindWindow-WindowTitle pic x(80).
      10  apiFindWindow-WindowTitle-type.
         15  xString  pic s9(4) comp-5 value +0008.
         15  xEmpty   pic s9(4) comp-5 value +0000.
         15  filler   pic s9(4) comp-5 value +0000.
         15  filler   pic x(4) value low-values.
```

Example:

The method takes a single parameter – the window title. Window titles are not case sensitive. When passed a window title, the function returns zero or the handle of the window that it finds.

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiFindWindow-Prop-Name to Sp2-OC-Prop-Name
Move apiFindWindow-Access-Type to Sp2-OC-Access-Type
Move apiFindWindow-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiFindWindow-Value-Type to Sp2-OC-Value-Type
Move “Window Title To Find” TO apiFindWindow-WindowTitle
Move apiFindWindow-WindowTitle to Sp2-OC-Parm-Value (1)
Move apiFindWindow-WindowTitle-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Either “0” or a number corresponding to the window handle of the window found, will be returned.

apiShowWindow (Method)

This method – when passed the handle of a window and a code for how to show the window, this method causes the specified window to be shown using the specified method. In general this can be used to restore a minimized application, show a hard to find window, etc. However, be aware that while the window will be shown. And depending on the setting you specify, should be given focus – it might not always have focus. Windows have what is called a Z order which determines which window is on top. There are actually two sets of Z order. The window at the top of the Z order has focus. A window can be at the top of the Z order and still not get focus if there are other windows that are known as “Top Most” windows, that are higher in the Z order. Look at the *grabFocus* method for a way to ensure that a window is at the top of the Z order of top most windows.

SP2-OCX-DEFINITION:

```
      01  api-apiShowWindow.
          05  apiShowWindow-prop-name
                       pic x(30) value "apiShowWindow".
          05  apiShowWindow-access-type
                       pic s9(4) comp-5 value +0001.
          05  apiShowWindow-parm-cnt
                       pic s9(4) comp-5 value +0002.
          05  apiShowWindow-value-type.
              10  xLong    pic s9(4) comp-5 value +0003.
              10  xEmpty   pic s9(4) comp-5 value +0000.
              10  filler   pic s9(4) comp-5 value +0000.
              10  filler   pic x(4) value low-values.
          05  apiShowWindow-parms.
              10  apiShowWindow-WindowHandle pic x(80).
              10  apiShowWindow-WindowHandle-type.
                  15  xLong    pic s9(4) comp-5 value +0003.
                  15  xEmpty   pic s9(4) comp-5 value +0000.
                  15  filler   pic s9(4) comp-5 value +0000.
                  15  filler   pic x(4) value low-values.
              10  apiShowWindow-ShowOption pic x(80).
              10  apiShowWindow-ShowOption-type.
                  15  xInteger pic s9(4) comp-5 value +0002.
                  15  xEmpty   pic s9(4) comp-5 value +0000.
                  15  filler   pic s9(4) comp-5 value +0000.
                  15  filler   pic x(4) value low-values.
```

ShowOption values:

Note: Some of these values duplicate native sp2 functionality. It is recommended that, when available, the native sp2 methods be used.

HIDE 0

Hides the window and activates another window.

SHOWNORMAL 1

Activates and displays a window. If the window is minimized or maximized, the system restores it to its original size and position.

SHOWMINIMIZED 2

Activates the window and displays it as a minimized window.

SHOWMAXIMIZED 3

Activates the window and displays it as a maximized window.

SHOWNOACTIVATE 4

Displays a window in its most recent size and position. This value is similar to SW_SHOWNORMAL, except the window is not activated.

SHOW 5

Activates the window and displays it in its current size and position.

MINIMIZE 6

Minimizes the specified window and activates the next top-level window in the Z order.

SHOWMINNOACTIVE 7

Displays the window as a minimized window. This value is similar to SHOWMINIMIZED, except the window is not activated.

SHOWNA 8

Displays the window in its current size and position. This value is similar to SHOW, except the window is not activated.

RESTORE 9

Activates and displays the window. If the window is minimized or maximized, the system restores it to its original size and position.

SHOWDEFAULT 10

Shows the window in its original creation state.

FORCEMINIMIZE 11 (XP/2000 Only)

Minimizes a window, even if the thread that owns the window is hung. This flag should only be used when minimizing windows from a different thread.

Example:

See CHECKACT.CBL

grabFocus (Method)

When passed a handle to a window, this method forces the window to the top of the Z Order of the top most windows and gives it focus. Since this may be different than the window hosting the ApiSimple control – you should use caution to not cause the window hosting the ApiSimple to grab focus back after using this method. An example is provided in the GRABFOC.CBL and DEMOFOC.CBL sample programs. DEMOFOC.CBL opens two windows then calls GRABFOC.CBL which forces the second of the windows to the front. You will note that when you run the program, while the second window is in the front now, the first has focus. Normally GRABFOC would be called from the window that needs focus – not from a different window. Sp2 is setting focus back on the first window after the call to GRABFOC because that is the window that was current when GRABFOC was called.

Note: When grabbing focus on a child window this function will bring the parent window to the front. If you need to work with a child window, set the property: topNotTopmost to true, and the child should get focus.

SP2-OCX-DEFINITION:

```
    01  api-grabFocus.
        05  grabFocus-prop-name
                      pic x(30) value "grabFocus".
        05  grabFocus-access-type
                      pic s9(4) comp-5 value +0001.
        05  grabFocus-parm-cnt
                      pic s9(4) comp-5 value +0001.
        05  grabFocus-value-type.
            10  xVoid    pic s9(4) comp-5 value +0024.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  grabFocus-parms.
            10  grabFocus-WindowHandle pic x(80).
            10  grabFocus-WindowHandle-type.
                15  xLong    pic s9(4) comp-5 value +0003.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

Example:

See DEMOFOC.CBL and GRABFOC.CBL. Note especially the way the window handle is constructed from an sp2-get-window-def call.

topNotTopmost (Property)

Set to true to cause the grabFocus method to use the predefined window wndTop instead of wndTopMost. The default is to use the topmost window which works in most cases. When a child window is involved, it is possible when wndTopMost is used, that the parent will be given focus. In this case, use the topNotTopmost property to indicate to ApiSimple to use wndTop and not wndTopmost.

SP2-OCX-DEFINITION:

```
    01  api-topNotTopmost.
       05  topNotTopmost-prop-name
                     pic x(30) value "topNotTopmost".
       05  topNotTopmost-access-type
                     pic s9(4) comp-5 value +0000.
       05  topNotTopmost-parm-cnt
                     pic s9(4) comp-5 value +0000.
       05  topNotTopmost-value-type.
           10  xBoolean pic s9(4) comp-5 value +0011.
           10  xEmpty   pic s9(4) comp-5 value +0000.
           10  filler   pic s9(4) comp-5 value +0000.
           10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move topNotTopmost-Prop-Name to Sp2-OC-Prop-Name
Move topNotTopmost-Access-Type to Sp2-OC-Access-Type
Move topNotTopmost-parm-Cnt to Sp2-OC-Parm-Cnt
Move topNotTopmost-Value-Type to Sp2-OC-Value-Type
Move “1” to sp2-oc-value
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Note: 1 = Boolean True, 0 = Boolean False

getMD5 (Method)

When passed a string of characters, this method returns the MD5 hash value for that string. This function is useful for password storage and validation. Instead of storing a users password, you store the MD5 value of their password. When they login, you compare the MD5 value of the password entered with that of the stored password. This makes it more difficult for your passwords to be compromised. Even if someone were to get the MD5 hash values, it is difficult to determine the password that created the MD5 hash. More information about MD5 can be found here:

http://www.rsasecurity.com/rsalabs/faq/3-6-6.html

SP2-OCX-DEFINITION:

```
      01  api-getMD5.
          05  getMD5-prop-name
                          pic x(30) value "getMD5".
          05  getMD5-access-type
                          pic s9(4) comp-5 value +0001.
          05  getMD5-parm-cnt
                          pic s9(4) comp-5 value +0001.
          05  getMD5-value-type.
              10  xString  pic s9(4) comp-5 value +0008.
              10  xEmpty   pic s9(4) comp-5 value +0000.
              10  filler   pic s9(4) comp-5 value +0000.
              10  filler   pic x(4) value low-values.
          05  getMD5-parms.
              10  getMD5-string pic x(80).
              10  getMD5-string-type.
                  15  xString  pic s9(4) comp-5 value +0008.
                  15  xEmpty   pic s9(4) comp-5 value +0000.
                  15  filler   pic s9(4) comp-5 value +0000.
                  15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getMD5-Prop-Name to Sp2-OC-Prop-Name
Move getMD5-Access-Type to Sp2-OC-Access-Type
Move getMD5-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getMD5-Value-Type to Sp2-OC-Value-Type
Move “Compute a hash for this string” TO getMD5-String
Move getMD5-String to Sp2-OC-Parm-Value (1)
Move getMD5-String-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

SP2-OC-VALUE will contain the 32 character hash value for the string passed.

apiDeleteFile (Method)

This method deletes the passed file name.

SP2-OCX-DEFINITION:

```
01  api-apiDeleteFile.
    05  apiDeleteFile-prop-name
                   pic x(30) value "apiDeleteFile".
    05  apiDeleteFile-access-type
                   pic s9(4) comp-5 value +0001.
    05  apiDeleteFile-parm-cnt
                   pic s9(4) comp-5 value +0001.
    05  apiDeleteFile-value-type.
        10  xString  pic s9(4) comp-5 value +0008.
        10  xEmpty   pic s9(4) comp-5 value +0000.
        10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
    05  apiDeleteFile-parms.
        10  apiDeleteFile-fileName pic x(80).
        10  apiDeleteFile-fileName-type.
            15  xString  pic s9(4) comp-5 value +0008.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiDeleteFile-Prop-Name to Sp2-OC-Prop-Name
Move apiDeleteFile-Access-Type to Sp2-OC-Access-Type
Move apiDeleteFile-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiDeleteFile-Value-Type to Sp2-OC-Value-Type
Move “C:\mydirectory\Myfile.dat” TO apiDeleteFile-fileName
Move apiDeleteFile-fileName to Sp2-OC-Parm-Value (1)
Move apiDeleteFile-fileName-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Sp2-OC-Value will return “Success” if the file was deleted or “Fail” if not. Failure could mean the file is not found, or that it was in use. A call to apiGetLastError might help determine what the problem is.

apiGetLastError (Method)

This method returns the status from the last API call that might update the last error value. You will need to lookup the error returned based on the last call made. http://msdn.microsoft.com is a good place to start.

SP2-OCX-DEFINITION

```
    01  api-apiGetLastError.
        05  apiGetLastError-prop-name
                      pic x(30) value "apiGetLastError".
        05  apiGetLastError-access-type
                      pic s9(4) comp-5 value +0001.
        05  apiGetLastError-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  apiGetLastError-value-type.
            10  xLong    pic s9(4) comp-5 value +0003.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiGetLastError-Prop-Name to Sp2-OC-Prop-Name
Move apiGetLastError-Access-Type to Sp2-OC-Access-Type
Move apiGetLastError-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiGetLastError-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return SP2-OC-Value will contain the numeric windows error code. Normally a zero is success.

apiDeleteFileWC (Method)

This method works similarly to apiDeleteFile, but instead of accepting just a file name it accepts a wildcard file name. Two parameters are passed – the first is the folder from which to delete the files and the second is the mask. For example C:\MYDATA in the folder name and *.DDD in the file name cause the method to delete all files as if DEL C:\MYDATA\*.DDD were entered. Note: This does not shell to a command prompt and enter the delete command – this actually reads through the directory and deletes files found that match the mask. Any of the traditional file name masks may be specified.

SP2-OCX-DEFINITION

```
    01  api-apiDeleteFileWC.
        05  apiDeleteFileWC-prop-name
                    pic x(30) value "apiDeleteFileWC".
        05  apiDeleteFileWC-access-type
                    pic s9(4) comp-5 value +0001.
        05  apiDeleteFileWC-parm-cnt
                    pic s9(4) comp-5 value +0002.
        05  apiDeleteFileWC-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  apiDeleteFileWC-parms.
            10  apiDeleteFileWC-folder pic x(80).
            10  apiDeleteFileWC-folder-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
            10  apiDeleteFileWC-mask pic x(80).
            10  apiDeleteFileWC-mask-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiDeleteFileWC-Prop-Name to Sp2-OC-Prop-Name
Move apiDeleteFileWC-Access-Type to Sp2-OC-Access-Type
Move apiDeleteFileWC-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiDeleteFileWC-Value-Type to Sp2-OC-Value-Type
Move “C:\MYDATA” TO apiDeleteFileWC-Folder
Move “*.NNN” TO apiDeleteFileWC-mask
Move apiDeleteFileWC-Folder to Sp2-OC-Parm-Value (1)
Move apiDeleteFileWC-Folder-Type to Sp2-OC-Parm-Type (1)
Move apiDeleteFileWC-mask to Sp2-OC-Parm-Value (2)
Move apiDeleteFileWC-mask-Type to Sp2-OC-Parm-Type (2)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

“Success” or “Fail” will be returned in Sp2-OC-value.

Note: Do not include a trailing “\” on the folder name.

killProcess (Method)

This method accepts a window title as a parameter, then kills the process that the window is associated with. Window titles are not case sensitive. The method attempts to cleanly end the chosen but if the process is not ended by the time an optional timeout occurs (Default 5 seconds if not specified) then the process is abruptly ended. The timeout in milliseconds is specified using a property – killTimeout – if the property is not set, 5000 milliseconds is used.

SP2-OCX-DEFINITION:

```
    01  api-killProcess.
        05  killProcess-prop-name
                       pic x(30) value "killProcess".
        05  killProcess-access-type
                       pic s9(4) comp-5 value +0001.
        05  killProcess-parm-cnt
                       pic s9(4) comp-5 value +0001.
        05  killProcess-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  killProcess-parms.
            10  killProcess-WindowTitle pic x(80).
            10  killProcess-WindowTitle-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

The **killTimeout** property is defined as follows:

```
     01  api-killTimeout.
         05  killTimeout-prop-name
                      pic x(30) value "killTimeout".
         05  killTimeout-access-type
                      pic s9(4) comp-5 value +0000.
         05  killTimeout-parm-cnt
                      pic s9(4) comp-5 value +0000.
         05  killTimeout-value-type.
              10  xLong    pic s9(4) comp-5 value +0003.
              10  xEmpty   pic s9(4) comp-5 value +0000.
              10  filler   pic s9(4) comp-5 value +0000.
              10  filler   pic x(4) value low-values.
```

Example:

This example kills a window with the title “To Be Killed” using a 15 second timeout.

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move killTimeout-Prop-Name to Sp2-OC-Prop-Name
Move killTimeout-Type to Sp2-OC-Access-Type
Move killTimeout-Parm-Cnt to Sp2-OC-Parm-Cnt
Move killTimeout-Value-Type to Sp2-OC-Value-Type
Move “15000” to SP2-OC-VALUE
CALL “SP2” using sp2-set-ocx sp2-ocx-parm

Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move killProcess-Prop-Name to Sp2-OC-Prop-Name
Move killProcess-Access-Type to Sp2-OC-Access-Type
Move killProcess-Parm-Cnt to Sp2-OC-Parm-Cnt
Move killProcess-Value-Type to Sp2-OC-Value-Type
Move “To Be Killed” To killProcess-WindowTitle
Move killProcess-WindowTitle to Sp2-OC-Parm-Value (1)
Move killProcess-WindowTitle-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

After the method completes, Sp2-OC-Value will contain one of the following:

Fail: Window not Found
Fail: Unable to get process id
Fail: Unknown
Fail: Termination Failed
Success: Clean Termination
Success: Terminated but Killed

Shell (Method)

The Shell method implements the ShellExecute API call. This api call has many functions. As you are aware, windows has file associations. One can click on a file of a particular extension and windows will launch the appropriate application for that file. This happens via the ShellExecute API. There are many uses for this feature. Since it launches applications via their association, it can handle the following – and more:

1. Starting the PDF viewer when only passing a PDF file name
   a. There is no need for you to know where the PDF viewer is installed
2. Launching the browser and taking the user to a web page
   a. This uses the file association of the default browser with any URL
3. Launching the word processor for a document
4. Launching the default email client when passed mailto:someone@somewhere.com
5. Starting an EXE of your choice

Read more about ShellExecute here:

http://msdn.microsoft.com/library/en-us/shellcc/platform/shell/reference/functions/shellexecute.asp

Calls to the method may be preceded by setting certain properties. At a minimum the “verb” property must be set.

verb (Property)

The “verb” property tells the Shell method which action you desire for ShellExecute. Normally this will be “open”, but all of the actions documented for ShellExecute are supported. The “Shell” method will assume “Open”, but the ShellWait requires that a verb be specified. The verbs (or operations) are:

edit
Launches an editor and opens the document for editing. If the name passed to the shell method is not a document file, the function will fail.

explore
Explores the folder specified by the parameter passed to the shell method.

find
Initiates a search starting from the specified directory (See startupFolder property)

open
Opens the file specified by the parameter passed to the shell method. The file can be an executable file, a document file, or a folder.

print
Prints the document file specified by the parameter passed to the shell method. If the parameter passed to the shell method is not a document file, the function will fail.

SP2-OCX-DEFINITION:

```
    01  api-verb.
        05  verb-prop-name
                        pic x(30) value "verb".
        05  verb-access-type
                        pic s9(4) comp-5 value +0000.
        05  verb-parm-cnt
                        pic s9(4) comp-5 value +0000.
        05  verb-value-type.
             10  xString  pic s9(4) comp-5 value +0008.
             10  xEmpty   pic s9(4) comp-5 value +0000.
             10  filler   pic s9(4) comp-5 value +0000.
             10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move verb-Prop-Name to Sp2-OC-Prop-Name
Move verb-Access-Type to Sp2-OC-Access-Type
Move verb-Parm-Cnt to Sp2-OC-Parm-Cnt
Move verb-Value-Type to Sp2-OC-Value-Type
Move “open” to SP2-OC-VALUE
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Optionally the **startupFolder** property may be set before using the Shell method. This will make the specified folder the current folder before the command is executed.

SP2-OCX-DEFINITION:

```
    01  api-startupFolder.
        05  startupFolder-prop-name
                       pic x(30) value "startupFolder".
        05  startupFolder-access-type
                       pic s9(4) comp-5 value +0000.
        05  startupFolder-parm-cnt
                       pic s9(4) comp-5 value +0000.
        05  startupFolder-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move startupFolder-Prop-Name to Sp2-OC-Prop-Name
Move startupFolder-Access-Type to Sp2-OC-Access-Type
Move startupFolder-Parm-Cnt to Sp2-OC-Parm-Cnt
Move startupFolder-Value-Type to Sp2-OC-Value-Type
Move “C:\MyFolder” to SP2-OC-VALUE
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

parms (Property)

Additional command line parameters may be specified before execution of the Shell method. These are specified in the parms property.

SP2-OCX-DEFINITION:

```
    01  api-parms.
        05  parms-prop-name
                       pic x(30) value "parms".
        05  parms-access-type
                       pic s9(4) comp-5 value +0000.
        05  parms-parm-cnt
                       pic s9(4) comp-5 value +0000.
        05  parms-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move parms-Prop-Name to Sp2-OC-Prop-Name
Move parms-Access-Type to Sp2-OC-Access-Type
Move parms-Parm-Cnt to Sp2-OC-Parm-Cnt
Move parms-Value-Type to Sp2-OC-Value-Type
Move “commandline-option” to SP2-OC-VALUE
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

showOption (Property) Available as of version 1.0.1.1

Set to an appropriate value for how the started processes from the Shell or ShellWait method should be displayed. The default is to show a normal window. The valid values are as follows:

HIDE 0
Hides the window and activates another window.

SHOWNORMAL 1
Activates and displays a window. If the window is minimized or maximized, the system restores it to its original size and position.

SHOWMINIMIZED 2
Activates the window and displays it as a minimized window.

SHOWMAXIMIZED 3
Activates the window and displays it as a maximized window.

SHOWNOACTIVATE 4
Displays a window in its most recent size and position. This value is similar to SW_SHOWNORMAL, except the window is not activated.

SHOW 5
Activates the window and displays it in its current size and position.

MINIMIZE 6
Minimizes the specified window and activates the next top-level window in the Z order.

SHOWMINNOACTIVE 7

Displays the window as a minimized window. This value is similar to SHOWMINIMIZED, except the window is not activated.

SHOWNA 8

Displays the window in its current size and position. This value is similar to SHOW, except the window is not activated.

RESTORE 9

Activates and displays the window. If the window is minimized or maximized, the system restores it to its original size and position.

SHOWDEFAULT 10

Shows the window in its original creation state.

FORCEMINIMIZE 11 (XP/2000 Only)

Minimizes a window, even if the thread that owns the window is hung. This flag should only be used when minimizing windows from a different thread.

SP2-OCX-DEFINITION:

```
    01  api-showOption.
        05  showOption-prop-name
                     pic x(80) value
           "showOption".
        05  showOption-access-type
                     pic s9(4) comp-5 value +0000.
        05  showOption-parm-cnt
                     pic s9(4) comp-5 value +0000.
        05  showOption-value-type.
           10  xInteger    pic s9(4) comp-5 value +0002.
           10  xEmpty      pic s9(4) comp-5 value +0000.
           10  filler      pic s9(4) comp-5 value +0000.
           10  filler      pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move showOption-Prop-Name to Sp2-OC-Prop-Name
Move showOption-Access-Type to Sp2-OC-Access-Type
Move showOption-Parm-Cnt to Sp2-OC-Parm-Cnt
Move showOption-Value-Type to Sp2-OC-Value-Type
Move “1” to SP2-OC-VALUE
```

```
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Shell (Method use)

The Shell method accepts a single parameter – the command or program on which to perform the action specified in the “verb” property. Execution of this method results in an immediate return and the process started runs asynchronously. Immediately after the process is started, the related properties, “verb”, “startupFolder” and “parms” are cleared. Subsequent invocations of the method require that you setup these values once again.

SP2-OCX-DEFINITION:

```
    01  api-Shell.
        05  Shell-prop-name
                      pic x(30) value "Shell".
        05  Shell-access-type
                      pic s9(4) comp-5 value +0001.
        05  Shell-parm-cnt
                      pic s9(4) comp-5 value +0001.
        05  Shell-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  Shell-parms.
            10  Shell-CommandLine pic x(80).
            10  Shell-CommandLine-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

Example (Starts default mail client and fills in To: address).

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move Shell-Prop-Name to Sp2-OC-Prop-Name
Move Shell-Access-Type to Sp2-OC-Access-Type
Move Shell-Parm-Cnt to Sp2-OC-Parm-Cnt
Move Shell-Value-Type to Sp2-OC-Value-Type
Move “mailto:techsupport@flexus.com” To Shell-CommandLine
Move Shell-Commandline to Sp2-OC-Parm-Value (1)
Move Shell-Commandline-Type to Sp2-OC-Parm-Type (1)
```

```
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The result of the call is returned in SP2-OC-VALUE and may be one of the following failures, or “Success” for success.

"Fail."
"The specified file is not found."
"The specified path is not found."
"The operating system denied access to the specified file."
"The file name association is incomplete or invalid."
"The DDE trans could not be completed--other DDE transactions being processed."
"The DDE transaction failed."
"The DDE transaction could not be completed because the request timed out."
"The specified dynamic-link library was not found."
"There is no application associated with the given file name extension."
“There was not enough memory to complete the operation."
"A sharing violation occurred."

ShellWait (Method)

This method is identical to the Shell method except it waits until the process is complete before returning to the program. The “verb” property ***MUST*** be set before using ShellWait. For documentation see “Shell” above.

retrieveVersionInfo (Method)

Windows programs, such as EXE’s, DLL’s and OCX files, may be marked with version information that is stored within the file as it is created. It may be useful to access this version information. This method accepts a file name as a parameter. Once executed, if “Success” is returned, follow up methods may be executed to retrieve various optionally available version information. If the file is not found, or a version control block is not available, “Fail” will be returned in SP2-OC-VALUE.

SP2-OCX-DEFINITION:

```
    01  api-retrieveVersionInfo.
        05  retrieveVersionInfo-prop-name
                      pic x(30) value "retrieveVersionInfo".
        05  retrieveVersionInfo-access-type
                      pic s9(4) comp-5 value +0001.
```

```
05  retrieveVersionInfo-parm-cnt
              pic s9(4) comp-5 value +0001.
05  retrieveVersionInfo-value-type.
    10  xString  pic s9(4) comp-5 value +0008.
    10  xEmpty   pic s9(4) comp-5 value +0000.
    10  filler   pic s9(4) comp-5 value +0000.
    10  filler   pic x(4) value low-values.
05  retrieveVersionInfo-parms.
    10  retrieveVersionInfo-fileName pic x(80).
    10  retrieveVersionInfo-fileName-type.
       15  xString  pic s9(4) comp-5 value +0008.
       15  xEmpty   pic s9(4) comp-5 value +0000.
       15  filler   pic s9(4) comp-5 value +0000.
       15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move retrieveVersionInfo-Prop-Name to Sp2-OC-Prop-Name
Move retrieveVersionInfo-Access-Type to Sp2-OC-Access-Type
Move retrieveVersionInfo-Parm-Cnt to Sp2-OC-Parm-Cnt
Move retrieveVersionInfo-Value-Type to Sp2-OC-Value-Type
Move “c:\whereitis\apisimp.ocx” To retrieveVersionInfo-fileName
Move retrieveVersionInfo-fileName to Sp2-OC-Parm-Value (1)
Move retrieveVersionInfo-fileName-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

After a successful call, the following methods will return the corresponding information in SP2-OC-VALUE when executed. Any of these may be executed in any order and any number of times once a successful retrieveVersionInfo method execution has occurred.

getComments
getCompanyName
getCreationTime
getFileDescription
getFileVersion
getFileVersionString
getInternalName
getLegalCopyright
getLegalTrademark

getOriginalFileName
getProductName
getProductVersion
getProductVersionString

getComments (Method)
After a successful execution of retrieveVersionInfo, this method returns the Comments stored in the file's version control block.

SP2-OCX-DEFINITION:

```
    01  api-getComments.
        05  getComments-prop-name
                      pic x(30) value "getComments".
        05  getComments-access-type
                      pic s9(4) comp-5 value +0001.
        05  getComments-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  getComments-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getComments-Prop-Name to Sp2-OC-Prop-Name
Move getComments-Access-Type to Sp2-OC-Access-Type
Move getComments-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getComments-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the comments – if available.

getCompanyName (Method)
After a successful execution of retrieveVersionInfo, this method returns the company name stored in the file's version control block.

SP2-OCX-DEFINITION:

```
    01  api-getCompanyName.
        05  getCompanyName-prop-name
                        pic x(30) value " getCompanyName".
        05  getCompanyName-access-type
                        pic s9(4) comp-5 value +0001.
        05  getCompanyName-parm-cnt
                        pic s9(4) comp-5 value +0000.
        05  getCompanyName-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getCompanyName-Prop-Name to Sp2-OC-Prop-Name
Move getCompanyName-Access-Type to Sp2-OC-Access-Type
Move getCompanyName-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getCompanyName-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the company name – if available.

getCreationTime (Method)

After a successful execution of retrieveVersionInfo, this method returns the file creation time stored in the file's version control block.

SP2-OCX-DEFINITION:

```
    01  api-getCreationTime.
        05  getCreationTime-prop-name
                        pic x(30) value "getCreationTime".
        05  getCreationTime-access-type
                        pic s9(4) comp-5 value +0001.
        05  getCreationTime-parm-cnt
                        pic s9(4) comp-5 value +0000.
        05  getCreationTime-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getCreationTime-Prop-Name to Sp2-OC-Prop-Name
Move getCreationTime-Access-Type to Sp2-OC-Access-Type
Move getCreationTime-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getCreationTime-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the creation time – if available.

Note: If no value is stored, January 1, 1601 is returned.

getFileDescription (Method)
After a successful execution of retrieveVersionInfo, this method returns the file creation time stored in the file’s version control block.

SP2-OCX-DEFINITION:

```
    01  api-getFileDescription.
        05  getFileDescription-prop-name
                     pic x(30) value "getFileDescription".
        05  getFileDescription-access-type
                     pic s9(4) comp-5 value +0001.
        05  getFileDescription-parm-cnt
                     pic s9(4) comp-5 value +0000.
        05  getFileDescription-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getFileDescription-Prop-Name to Sp2-OC-Prop-Name
Move getFileDescription-Access-Type to Sp2-OC-Access-Type
Move getFileDescription-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getFileDescription-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the file description – if available.

getFileVersion (Method)
After a successful execution of retrieveVersionInfo, this method returns the file version stored in the file's version control block.

SP2-OCX-DEFINITION:

```
01  api-getFileVersion.
    05  getFileVersion-prop-name
                      pic x(30) value "getFileVersion".
    05  getFileVersion-access-type
                      pic s9(4) comp-5 value +0001.
    05  getFileVersion-parm-cnt
                      pic s9(4) comp-5 value +0000.
    05  getFileVersion-value-type.
         10  xString  pic s9(4) comp-5 value +0008.
         10  xEmpty   pic s9(4) comp-5 value +0000.
         10  filler   pic s9(4) comp-5 value +0000.
         10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getFileVersion-Prop-Name to Sp2-OC-Prop-Name
Move getFileVersion-Access-Type to Sp2-OC-Access-Type
Move getFileVersion-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getFileVersion-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the file version – if available.

getFileVersionString (Method)
After a successful execution of retrieveVersionInfo, this method returns the file version as a string stored in the file's version control block.

SP2-OCX-DEFINITION:

```
     01  api-getFileVersionString.
         05  getFileVersionString-prop-name
                       pic x(30) value "getFileVersionString".
         05  getFileVersionString-access-type
                       pic s9(4) comp-5 value +0001.
         05  getFileVersionString-parm-cnt
                       pic s9(4) comp-5 value +0000.
         05  getFileVersionString-value-type.
             10  xString  pic s9(4) comp-5 value +0008.
             10  xEmpty   pic s9(4) comp-5 value +0000.
             10  filler   pic s9(4) comp-5 value +0000.
             10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getFileVersionString-Prop-Name to Sp2-OC-Prop-Name
Move getFileVersionString-Access-Type to Sp2-OC-Access-Type
Move getFileVersionString-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getFileVersionString-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the file version as a string – if available.

getInternalName (Method)
After a successful execution of retrieveVersionInfo, this method returns the internal name stored in the file’s version control block.

SP2-OCX-DEFINITION:

```
     01  api-getInternalName.
         05  getInternalName-prop-name
                       pic x(30) value "getInternalName".
         05  getInternalName-access-type
                       pic s9(4) comp-5 value +0001.
         05  getInternalName-parm-cnt
                       pic s9(4) comp-5 value +0000.
         05  getInternalName-value-type.
             10  xString  pic s9(4) comp-5 value +0008.
```

```
          10  xEmpty   pic s9(4) comp-5 value +0000.
          10  filler   pic s9(4) comp-5 value +0000.
          10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getInternalName-Prop-Name to Sp2-OC-Prop-Name
Move getInternalName-Access-Type to Sp2-OC-Access-Type
Move getInternalName-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getInternalName-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the internal name – if available.

getLegalCopyright (Method)
After a successful execution of retrieveVersionInfo, this method returns the legal copyright stored in the file's version control block.

SP2-OCX-DEFINITION:

```
   01  api-getLegalCopyright.
       05  getLegalCopyright-prop-name
                     pic x(30) value "getLegalCopyright".
       05  getLegalCopyright-access-type
                     pic s9(4) comp-5 value +0001.
       05  getLegalCopyright-parm-cnt
                     pic s9(4) comp-5 value +0000.
       05  getLegalCopyright-value-type.
           10  xString  pic s9(4) comp-5 value +0008.
           10  xEmpty   pic s9(4) comp-5 value +0000.
           10  filler   pic s9(4) comp-5 value +0000.
           10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getLegalCopyright-Prop-Name to Sp2-OC-Prop-Name
Move getLegalCopyright-Access-Type to Sp2-OC-Access-Type
Move getLegalCopyright-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getLegalCopyright-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the legal copyright – if available.

Note: The fact that a file doesn’t have a legal copyright does not mean it is not copyrighted – it only means that the copyright information is not recorded in the files version control block.

getLegalTrademark (Method)

After a successful execution of retrieveVersionInfo, this method returns the legal trademark stored in the file’s version control block.

SP2-OCX-DEFINITION:

```
01  api-getLegalTrademark.
   05  getLegalTrademark-prop-name
                 pic x(30) value "getLegalTrademark".
   05  getLegalTrademark-access-type
                 pic s9(4) comp-5 value +0001.
   05  getLegalTrademark-parm-cnt
                 pic s9(4) comp-5 value +0000.
   05  getLegalTrademark-value-type.
       10  xString  pic s9(4) comp-5 value +0008.
       10  xEmpty   pic s9(4) comp-5 value +0000.
       10  filler   pic s9(4) comp-5 value +0000.
       10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getLegalTrademark-Prop-Name to Sp2-OC-Prop-Name
Move getLegalTrademark-Access-Type to Sp2-OC-Access-Type
Move getLegalTrademark-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getLegalTrademark-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the legal tradmark – if available.

Note: The fact that a file doesn't have a legal trademark does not mean it is not trademarked – it only means that the trademark information is not recorded in the files version control block.

getOriginalFileName (Method)
After a successful execution of retrieveVersionInfo, this method returns the original file name stored in the file's version control block.

SP2-OCX-DEFINITION:

```
       01  api-getOriginalFileName.
           05  getOriginalFileName-prop-name
                         pic x(30) value "getOriginalFileName".
           05  getOriginalFileName-access-type
                         pic s9(4) comp-5 value +0001.
           05  getOriginalFileName-parm-cnt
                         pic s9(4) comp-5 value +0000.
           05  getOriginalFileName-value-type.
               10  xString  pic s9(4) comp-5 value +0008.
               10  xEmpty   pic s9(4) comp-5 value +0000.
               10  filler   pic s9(4) comp-5 value +0000.
               10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getOriginalFileName-Prop-Name to Sp2-OC-Prop-Name
Move getOriginalFileName-Access-Type to Sp2-OC-Access-Type
Move getOriginalFileName-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getOriginalFileName-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the original file name – if available.

getProductName (Method)
After a successful execution of retrieveVersionInfo, this method returns the product name stored in the file's version control block.

SP2-OCX-DEFINITION:

```
       01  api-getProductName.
```

```
      05  getProductName-prop-name
                      pic x(30) value "getProductName".
      05  getProductName-access-type
                      pic s9(4) comp-5 value +0001.
      05  getProductName-parm-cnt
                      pic s9(4) comp-5 value +0000.
      05  getProductName-value-type.
          10  xString  pic s9(4) comp-5 value +0008.
          10  xEmpty   pic s9(4) comp-5 value +0000.
          10  filler   pic s9(4) comp-5 value +0000.
          10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getProductName-Prop-Name to Sp2-OC-Prop-Name
Move getProductName-Access-Type to Sp2-OC-Access-Type
Move getProductName-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getProductName-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the product name – if available.

getProductVersion (Method)
After a successful execution of retrieveVersionInfo, this method returns the product version stored in the file’s version control block.

SP2-OCX-DEFINITION:

```
   01  api-getProductVersion.
      05  getProductVersion-prop-name
                      pic x(30) value "getProductVersion".
      05  getProductVersion-access-type
                      pic s9(4) comp-5 value +0001.
      05  getProductVersion-parm-cnt
                      pic s9(4) comp-5 value +0000.
      05  getProductVersion-value-type.
          10  xString  pic s9(4) comp-5 value +0008.
          10  xEmpty   pic s9(4) comp-5 value +0000.
          10  filler   pic s9(4) comp-5 value +0000.
          10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getProductVersion-Prop-Name to Sp2-OC-Prop-Name
Move getProductVersion-Access-Type to Sp2-OC-Access-Type
Move getProductVersion-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getProductVersion-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the product version – if available.

getProductVersionString (Method)

After a successful execution of retrieveVersionInfo, this method returns the product version as a string stored in the file’s version control block.

SP2-OCX-DEFINITION:

```
    01  api-getProductVersionString.
        05  getProductVersionString-prop-name
                      pic x(30) value "getProductVersionString".
        05  getProductVersionString-access-type
                      pic s9(4) comp-5 value +0001.
        05  getProductVersionString-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  getProductVersionString-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getProductVersionString-Prop-Name to Sp2-OC-Prop-Name
Move getProductVersionString-Access-Type to Sp2-OC-Access-Type
Move getProductVersionString-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getProductVersionString-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
```

```
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the product version – if available.

apiFindFirstFile (Method)
This method is similar to the various “Check File Exist” features of many COBOL implementations. Not all information returned from this function is currently available, but after making a successful call to apiFindFirstFile, many properties may be interrogated to find out specific information about the file sought.

SP2-OCX-DEFINITION:

```
    01  api-apiFindFirstFile.
        05  apiFindFirstFile-prop-name
                      pic x(30) value "apiFindFirstFile".
        05  apiFindFirstFile-access-type
                      pic s9(4) comp-5 value +0001.
        05  apiFindFirstFile-parm-cnt
                      pic s9(4) comp-5 value +0001.
        05  apiFindFirstFile-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  apiFindFirstFile-parms.
            10  apiFindFirstFile-fileName pic x(80).
            10  apiFindFirstFile-fileName-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiFindFirstFile-Prop-Name to Sp2-OC-Prop-Name
Move apiFindFirstFile-Access-Type to Sp2-OC-Access-Type
Move apiFindFirstFile-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiFindFirstFile-Value-Type to Sp2-OC-Value-Type
Move “c:\myfiles\myfile.dat” To apiFindFirstFile-fileName
Move apiFindFirstFile-fileName to Sp2-OC-Parm-Value (1)
Move apiFindFirstFile-fileName-Type to Sp2-OC-Parm-Type (1)
```

```
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

If the file is found, "Success" is returned in SP2-OC-VALUE, otherwise "Fail" is returned. If a file is found, three properties are currently available for interrogation. (Please see the note about enhancements at the end of this document).

DwHighDateAndTime (Property)
This property value is set after a successful invocation of apiFindFirstFile. It contains what Microsoft calls the "high" portion of the last date and time the file was written. This is part of the date that is translated into the date you see when you a directory of the file. It is provided her for it's use in file transfer using thin client and ApiSimple. Using this and DwLowDateAndTime one may set the timestamp (using the setLastWriteTime method) so that the timestamps on the file transferred matches the source file.

SP2-OCX-DEFINITION:

```
    01  api-DwHighDateAndTime.
        05  DwHighDateAndTime-prop-name
                      pic x(30) value "DwHighDateAndTime".
        05  DwHighDateAndTime-access-type
                      pic s9(4) comp-5 value +0000.
        05  DwHighDateAndTime-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  DwHighDateAndTime-value-type.
            10  xLong    pic s9(4) comp-5 value +0003.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move DwHighDateAndTime-Prop-Name to Sp2-OC-Prop-Name
Move DwHighDateAndTime-Access-Type to Sp2-OC-Access-Type
Move DwHighDateAndTime-Parm-Cnt to Sp2-OC-Parm-Cnt
Move DwHighDateAndTime-Value-Type to Sp2-OC-Value-Type
CALL "SP2" using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

Since SP2-OC-VALUE is an alphanumeric field, you will need to convert the returned value to a number.

DwLowDateAndTime (Property)
This property value is set after a successful invocation of apiFindFirstFile. It contains what Microsoft calls the “low” portion of the last date and time the file was written. This is part of the date that is translated into the date you see when you a directory of the file. It is provided her for it’s use in file transfer using thin client and ApiSimple. Using this and DwLowDateAndTime one may set the timestamp (using the setLastWriteTime method) so that the timestamps on the file transferred matches the source file.

SP2-OCX-DEFINITION:

```
    01  api-DwLowDateAndTime.
        05  DwLowDateAndTime-prop-name
                      pic x(30) value "DwLowDateAndTime".
        05  DwLowDateAndTime-access-type
                      pic s9(4) comp-5 value +0000.
        05  DwLowDateAndTime-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  DwLowDateAndTime-value-type.
            10  xLong    pic s9(4) comp-5 value +0003.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move DwLowDateAndTime-Prop-Name to Sp2-OC-Prop-Name
Move DwLowDateAndTime-Access-Type to Sp2-OC-Access-Type
Move DwLowDateAndTime-Parm-Cnt to Sp2-OC-Parm-Cnt
Move DwLowDateAndTime-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

Since SP2-OC-VALUE is an alphanumeric field, you will need to convert the returned value to a number.

FilesizeLow (Property)
This property value is set after a successful invocation of apiFindFirstFile. It contains what Microsoft calls the “low” portion of the file size. For most files this is adequate for determining the actual file size.

SP2-OCX-DEFINITION:

```
    01  api-filesizeLow.
        05  filesizeLow-prop-name
                         pic x(30) value "filesizeLow".
        05  filesizeLow-access-type
                         pic s9(4) comp-5 value +0000.
        05  filesizeLow-parm-cnt
                         pic s9(4) comp-5 value +0000.
        05  filesizeLow-value-type.
            10  xLong    pic s9(4) comp-5 value +0003.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move filesizeLow-Prop-Name to Sp2-OC-Prop-Name
Move filesizeLow-Access-Type to Sp2-OC-Access-Type
Move filesizeLow-parm-Cnt to Sp2-OC-Parm-Cnt
Move filesizeLow-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

Since SP2-OC-VALUE is an alphanumeric field, you will need to convert the returned value to a number.

Read more about these file times and sizes here:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/fileio/base/win32_find_data_str.asp

apiGetCurrentDirectory (Method) Available as of version 1.0.1.1

Returns the current directory. Ie: If C:\Myfolder is the current directory, then C:\Myfolder will be returned.

SP2-OCX-DEFINITION:

```
     * Name Shortened - was:
     * apiGetCurrentDirectory
      01  api-apiGtCrrntDrctry.
          05  apiGtCrrntDrctry-prop-name
                        pic x(80) value
              "apiGetCurrentDirectory".
          05  apiGtCrrntDrctry-access-type
                        pic s9(4) comp-5 value +0001.
          05  apiGtCrrntDrctry-parm-cnt
                        pic s9(4) comp-5 value +0000.
          05  apiGtCrrntDrctry-value-type.
              10  xString     pic s9(4) comp-5 value +0008.
              10  xEmpty      pic s9(4) comp-5 value +0000.
              10  filler      pic s9(4) comp-5 value +0000.
              10  filler      pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiGtCrrntDrctry-Prop-Name to Sp2-OC-Prop-Name
Move apiGtCrrntDrctry-Access-Type to Sp2-OC-Access-Type
Move apiGtCrrntDrctry-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiGtCrrntDrctry-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the current directory.

getVersion (Method)
This method returns the internal version of ApiSimple. As the control is enhanced and new methods and properties are added you may want to deploy a new version. To ensure that the proper version is running before you invoke the new methods or make use of the new properties, you can make a call to get the version.

SP2-OCX-DEFINITION:

```
      01  api-getVersion.
          05  getVersion-prop-name
                        pic x(30) value "getVersion".
          05  getVersion-access-type
                        pic s9(4) comp-5 value +0001.
          05  getVersion-parm-cnt
                        pic s9(4) comp-5 value +0000.
          05  getVersion-value-type.
              10  xString  pic s9(4) comp-5 value +0008.
              10  xEmpty   pic s9(4) comp-5 value +0000.
              10  filler   pic s9(4) comp-5 value +0000.
              10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move getVersion-Prop-Name to Sp2-OC-Prop-Name
Move getVersion-Access-Type to Sp2-OC-Access-Type
Move getVersion-Parm-Cnt to Sp2-OC-Parm-Cnt
Move getVersion-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return, SP2-OC-VALUE will contain the version of ApiSimple that is in use.

Shared Memory Access

ApiSimple supports a very basic shared memory feature. This allows separately running EXE’s to share an area of memory. ApiSimple has methods to establish this area with a name and size, and then to read from and write to the memory area. The following methods are used to support this feature:

mapMemory
writeMemory
readMemory
releaseMemory

There is also one associated property, myMemory.

The sequence of events is as follows:

First – you must map some memory to be used. Each instance of ApiSimple may only reference a single area. You assign the area a name and a size using parameters of the mapMemory method. After that, you may read from and write to this memory area. When you are finished with the area, use the releaseMemory method to free the memory. Application A can map the memory. When application B maps memory with the same name, it will reference the same memory referenced my application A. Application A may even end and the memory will remain so long as a program has a reference to it. When all programs end, the memory will be automatically released, but it is better to make the releaseMemory method call to ensure no memory is left over.

mapMemory (Method)
This method creates the shared memory if it does not already exist, or maps to the existing memory if it has already been created.

SP2-OCX-DEFINITION:

```
   01  api-mapMemory.
       05  mapMemory-prop-name
                    pic x(30) value "mapMemory".
       05  mapMemory-access-type
                    pic s9(4) comp-5 value +0001.
       05  mapMemory-parm-cnt
                    pic s9(4) comp-5 value +0002.
       05  mapMemory-value-type.
           10  xString  pic s9(4) comp-5 value +0008.
           10  xEmpty   pic s9(4) comp-5 value +0000.
           10  filler   pic s9(4) comp-5 value +0000.
           10  filler   pic x(4) value low-values.
       05  mapMemory-parms.
           10  mapMemory-Name pic x(80).
           10  mapMemory-Name-type.
               15  xString  pic s9(4) comp-5 value +0008.
               15  xEmpty   pic s9(4) comp-5 value +0000.
               15  filler   pic s9(4) comp-5 value +0000.
               15  filler   pic x(4) value low-values.
           10  mapMemory-Size pic x(80).
           10  mapMemory-Size-type.
               15  xLong    pic s9(4) comp-5 value +0003.
               15  xEmpty   pic s9(4) comp-5 value +0000.
               15  filler   pic s9(4) comp-5 value +0000.
               15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move mapMemory-Prop-Name to Sp2-OC-Prop-Name
Move mapMemory-Access-Type to Sp2-OC-Access-Type
Move mapMemory-Parm-Cnt to Sp2-OC-Parm-Cnt
Move mapMemory-Value-Type to Sp2-OC-Value-Type
Move “Whatever” TO mapMemory-Name
Move “1024” TO mapMemory-Size
Move mapMemory-Name to Sp2-OC-Parm-Value (1)
Move mapMemory-Name-Type to Sp2-OC-Parm-Type (1)
Move mapMemory-Size to Sp2-OC-Parm-Value (2)
Move mapMemory-Size-Type to Sp2-OC-Parm-Type (2)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return SP2-OC-VALUE will contain one of the following:

Success: New memory mapped
Success: Existing memory mapped
Fail

writeMemory (Method)
This method accepts two parameters. The first is the data to be written to the memory and the second is the size.

NOTE about how sp2 passes data to a control. Sp2 must determine the length of the data. This occurs by sp2 searching from the end of the data area for the first character that is not a space and is not a low-value. If you need to pass trailing spaces for this property or a trailing set of low-value characters, append the data with an “X” or some other character. This will not cause any harm because the writeMemory method is passed a length – this length will not include the trailing “X” character.

SP2-OCX-DEFINITION:

```
    01  api-writeMemory.
        05  writeMemory-prop-name
                    pic x(30) value "writeMemory".
        05  writeMemory-access-type
                    pic s9(4) comp-5 value +0001.
        05  writeMemory-parm-cnt
                    pic s9(4) comp-5 value +0002.
        05  writeMemory-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  writeMemory-parms.
            10  writeMemory-Data pic x(80).
        10  writeMemory-Data-type.
            15  xString  pic s9(4) comp-5 value +0008.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
        10  writeMemory-Length pic x(80).
        10  writeMemory-Length-type.
            15  xLong    pic s9(4) comp-5 value +0003.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move writeMemory-Prop-Name to Sp2-OC-Prop-Name
Move writeMemory-Access-Type to Sp2-OC-Access-Type
Move writeMemory-Parm-Cnt to Sp2-OC-Parm-Cnt
Move writeMemory-Value-Type to Sp2-OC-Value-Type
Move Data-area TO writeMemory-Data
Move “1024” TO writeMemory-Length
Move writeMemory-Data to Sp2-OC-Parm-Value (1)
Move writeMemory-Data-Type to Sp2-OC-Parm-Type (1)
Move writeMemory-Length to Sp2-OC-Parm-Value (2)
Move writeMemory-Length-Type to Sp2-OC-Parm-Type (2)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return SP2-OC-VALUE will contain one of the following:

Success
Fail: Memory Locked or Unavailable

The memory is locked while another process is accessing it, however, you should not see this using the current implementation as ApiSimple only locks the memory for a short time on a write, and waits for it to become available on a read. If you do get the Fail message, it is probably because the memory block has been released or was never mapped.

readMemory (Method)

This method reads the memory block and returns the status of the read. The actual data returned is stored in the myMemory property and must be retrieved via that property.

SP2-OCX-DEFINITION:

```
    01  api-readMemory.
        05  readMemory-prop-name
                     pic x(30) value "readMemory".
        05  readMemory-access-type
                     pic s9(4) comp-5 value +0001.
        05  readMemory-parm-cnt
                     pic s9(4) comp-5 value +0000.
        05  readMemory-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move readMemory-Prop-Name to Sp2-OC-Prop-Name
Move readMemory-Access-Type to Sp2-OC-Access-Type
Move readMemory-Parm-Cnt to Sp2-OC-Parm-Cnt
Move readMemory-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return SP2-OC-VALUE will contain one of the following:

Success
Fail: Memory Locked or Unavailable

The memory is locked while another process is accessing it, however, you should not see this using the current implementation as ApiSimple only locks the memory for a short time on a write, and waits for it to become available on a read. If you do get the Fail message, it is probably because the memory block has been released or was never mapped.

myMemory (Property)
This property contains the data read from memory the readMemory method.

SP2-OCX-DEFINITION:

```
   01  api-myMemory.
       05  myMemory-prop-name
                    pic x(30) value "myMemory".
       05  myMemory-access-type
                    pic s9(4) comp-5 value +0000.
       05  myMemory-parm-cnt
                    pic s9(4) comp-5 value +0000.
       05  myMemory-value-type.
           10  xString  pic s9(4) comp-5 value +0008.
           10  xEmpty   pic s9(4) comp-5 value +0000.
           10  filler   pic s9(4) comp-5 value +0000.
           10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move myMemory-Prop-Name to Sp2-OC-Prop-Name
Move myMemory-Access-Type to Sp2-OC-Access-Type
Move myMemory-Parm-Cnt to Sp2-OC-Parm-Cnt
Move myMemory-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE.

FILE and COM PORT I/O

The remaining functions of ApiSimple serve to handle bytestream file I/O and asynchronous communication. Not every aspect of such I/O is supported, but a basic set is available.

File I/O

The file I/O options are useful for reading and writing files in a stream of bytes. Files are read and written based on an offset and length, and not as “records”. These functions are quite useful in a thin client environment where you might need to read or create files on the client from your thin client server, and you don’t desire to, or it is inappropriate to copy the files. These functions have been used to create a file transfer system for thin client that includes progress indication for large files – unlike “Copy-File-To-Client” and “Copy-File-To-Server” calls which offer no progress indication.

There are two methods for opening a file, one for input and the other for output.

apiCreateFile (Method)
This method creates the specified file opening it output. If the file already exists, it is overwritten. A single parameter is passed – the file name. The file handle is held internally in the ActiveX control.

CAUTION: Only a single file or com port can be opened in an instance of the ApiSimple control. This is for simplicity. If you require another file or com port to be open concurrently, simply use another copy of the control – it can even be on the same sp2 panel.

SP2-OCX-DEFINITION:

```
    01  api-apiCreateFile.
        05  apiCreateFile-prop-name
                       pic x(30) value "apiCreateFile".
        05  apiCreateFile-access-type
                       pic s9(4) comp-5 value +0001.
        05  apiCreateFile-parm-cnt
                       pic s9(4) comp-5 value +0001.
        05  apiCreateFile-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  apiCreateFile-parms.
            10  apiCreateFile-fileName pic x(80).
            10  apiCreateFile-fileName-type.
                15  xString  pic s9(4) comp-5 value +0008.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiCreateFile-Prop-Name to Sp2-OC-Prop-Name
Move apiCreateFile-Access-Type to Sp2-OC-Access-Type
Move apiCreateFile-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiCreateFile-Value-Type to Sp2-OC-Value-Type
Move “c:\myfiles\myfile.dat” To apiCreateFile-fileName
Move apiCreateFile-fileName to Sp2-OC-Parm-Value (1)
Move apiCreateFile-fileName-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

If the file is created and open for writing, “Success” is returned in SP2-OC-VALUE, otherwise “Fail” is returned. The reason for any failure can be determined by using the apiGetLastError method.

NOTE: Don’t forget to close the file when you are finished using the apiCloseFile method.

apiOpenRead (Method)

This method opens a file for input. A single parameter is passed – the file name. The file handle is held internally in the ActiveX control.

CAUTION: Only a single file or com port can be opened in an instance of the ApiSimple control. This is for simplicity. If you require another file or com port to be open concurrently, simply use another copy of the control – it can even be on the same sp2 panel.

SP2-OCX-DEFINITION:

```
   01  api-apiOpenRead.
       05  apiOpenRead-prop-name
                     pic x(30) value "apiOpenRead".
       05  apiOpenRead-access-type
                     pic s9(4) comp-5 value +0001.
       05  apiOpenRead-parm-cnt
                     pic s9(4) comp-5 value +0001.
       05  apiOpenRead-value-type.
           10  xString  pic s9(4) comp-5 value +0008.
           10  xEmpty   pic s9(4) comp-5 value +0000.
           10  filler   pic s9(4) comp-5 value +0000.
           10  filler   pic x(4) value low-values.
       05  apiOpenRead-parms.
           10  apiOpenRead-fileName pic x(80).
           10  apiOpenRead-fileName-type.
               15  xString  pic s9(4) comp-5 value +0008.
               15  xEmpty   pic s9(4) comp-5 value +0000.
               15  filler   pic s9(4) comp-5 value +0000.
```

```
              15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiOpenRead-Prop-Name to Sp2-OC-Prop-Name
Move apiOpenRead-Access-Type to Sp2-OC-Access-Type
Move apiOpenRead-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiOpenRead-Value-Type to Sp2-OC-Value-Type
Move “c:\myfiles\myfile.dat” To apiOpenRead-fileName
Move apiOpenRead-fileName to Sp2-OC-Parm-Value (1)
Move apiOpenRead-fileName-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

If the file is open for reading, “Success” is returned in SP2-OC-VALUE, otherwise “Fail” is returned. The reason for any failure can be determined by using the apiGetLastError method.

NOTE: Don’t forget to close the file when you are finished using the apiCloseFile method.

apiReadData (Method)
This method reads data from a file opened with apiOpenRead. You pass the method the number of bytes to read. The data is read into the myData property where it can be retrieved. The data is read from the file starting at the last point in the file where data was previously read. The returned value, is SP2-OC-VALUE is the actual number of bytes read.

SP2-OCX-DEFINITION

```
    01  api-apiReadData.
        05  apiReadData-prop-name
                      pic x(30) value "apiReadData".
        05  apiReadData-access-type
                      pic s9(4) comp-5 value +0001.
        05  apiReadData-parm-cnt
                      pic s9(4) comp-5 value +0001.
        05  apiReadData-value-type.
            10  xLong    pic s9(4) comp-5 value +0003.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
        05  apiReadData-parms.
            10  apiReadData-bytesToRead pic x(80).
            10  apiReadData-bytesToRead-type.
                15  xLong    pic s9(4) comp-5 value +0003.
                15  xEmpty   pic s9(4) comp-5 value +0000.
                15  filler   pic s9(4) comp-5 value +0000.
                15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiReadData-Prop-Name to Sp2-OC-Prop-Name
Move apiReadData-Access-Type to Sp2-OC-Access-Type
Move apiReadData-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiReadData-Value-Type to Sp2-OC-Value-Type
Move “1500” To apiReadData-bytesToRead
Move apiReadData-bytesToRead to Sp2-OC-Parm-Value (1)
Move apiReadData-bytesToRead-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

NOTE: the maximum number of bytes that can be read at one time is 8192. Retrieve the data read using the myData property.

myData (property)
This property contains data read from a file or com port. It’s max size is 8192. Use the length returned from the read operation to get at the data.
NOTE: You may want to use the USR-OCX-PARMS definition included in USROCX.CPY since the parms are defined to be a larger size than the default 80 bytes of sp2ocx.cpy.

SP2-OCX-DEFINITION:

```
    01  api-myData.
        05  myData-prop-name
                    pic x(30) value "myData".
        05  myData-access-type
                    pic s9(4) comp-5 value +0000.
        05  myData-parm-cnt
                    pic s9(4) comp-5 value +0000.
        05  myData-value-type.
```

```
         10  xString  pic s9(4) comp-5 value +0008.
         10  xEmpty   pic s9(4) comp-5 value +0000.
         10  filler   pic s9(4) comp-5 value +0000.
         10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move myData-Prop-Name to Sp2-OC-Prop-Name
Move myData-Access-Type to Sp2-OC-Access-Type
Move myData-Parm-Cnt to Sp2-OC-Parm-Cnt
Move myData-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

apiWriteFileX (Method)
This method writes data to the file opened with the apiCreateFile method. This method must be preceded by setting the data to be written in the myDataBuffer property. After placing the data to be written in the myDataBuffer property, use this method, passing the number of bytes to write as the single parameter.

See Note about the myDataBuffer property and using trailing low-values or spaces.

SP2-OCX-DEFINITION

```
01  api-apiWriteFileX.
    05  apiWriteFileX-prop-name
                      pic x(30) value "apiWriteFileX".
    05  apiWriteFileX-access-type
                      pic s9(4) comp-5 value +0001.
    05  apiWriteFileX-parm-cnt
                      pic s9(4) comp-5 value +0001.
    05  apiWriteFileX-value-type.
         10  xLong    pic s9(4) comp-5 value +0003.
         10  xEmpty   pic s9(4) comp-5 value +0000.
         10  filler   pic s9(4) comp-5 value +0000.
         10  filler   pic x(4) value low-values.
    05  apiWriteFileX-parms.
         10  apiWriteFileX-bytes pic x(80).
         10  apiWriteFileX-bytes-type.
             15  xLong    pic s9(4) comp-5 value +0003.
```

```
                  15  xEmpty   pic s9(4) comp-5 value +0000.
                  15  filler   pic s9(4) comp-5 value +0000.
                  15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiWriteFileX-Prop-Name to Sp2-OC-Prop-Name
Move apiWriteFileX-Access-Type to Sp2-OC-Access-Type
Move apiWriteFileX-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiWriteFileX-Value-Type to Sp2-OC-Value-Type
Move “1500” To apiWriteFileX-bytes
Move apiWriteFileX-bytes to Sp2-OC-Parm-Value (1)
Move apiWriteFileX-bytes-Type to Sp2-OC-Parm-Type (1)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Upon return SP2-OC-VALUE will contain the number of bytes actually written to the file.

myDataBuffer (property)
This property contains the data to be written to the file using the apiWriteFileX method. Set this property with the data before writing. You may want to use the modified copybook in USROCX.CPY which contains bigger paramenters.

NOTE about how sp2 passes data to a control. Sp2 must determine the length of the data. This occurs by sp2 searching from the end of the data area for the first character that is not a space and is not a low-value. If you need to pass trailing spaces for this property or a trailing set of low-value characters, append the data with an “X” or some other character. This will not cause any harm because the apiWriteFileX is passed a length – this length will not include the trailing “X” character.

NOTE for thin client users. Do NOT Set sp2-oc-options = 2 for this property call. While ApiSimple could have passed data to the apiWriteFileX via a parm for that method using a separate property is more efficient. Since a return value from apiWriteFileX is desired – the number of bytes written – using the “2” in sp2-oc-options causes thin client to wait until the server receives the response before continuing. This requires a ROUND TRIP of the entire data area. Using a separate property and NOT specifying sp2-oc-options = 2, means that thin client does not need to transmit the data both directions – thus saving time and bandwidth.

SP2-OCX-DEFINITION:

```
    01  api-myDataBuffer.
        05  myDataBuffer-prop-name
                          pic x(30) value "myDataBuffer".
        05  myDataBuffer-access-type
                          pic s9(4) comp-5 value +0000.
        05  myDataBuffer-parm-cnt
                          pic s9(4) comp-5 value +0000.
        05  myDataBuffer-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

The maximum size of this property is whatever sp2 can pass. Previous versions of the ActiveX support were limited to 8192 bytes, but the newer version supports large data blocks.

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move myDataBuffer-Prop-Name to Sp2-OC-Prop-Name
Move myDataBuffer-Access-Type to Sp2-OC-Access-Type
Move myDataBuffer-parm-Cnt to Sp2-OC-Parm-Cnt
Move myData-Value-Type to Sp2-OC-Value-Type
Move “The data to be written” to sp2-oc-value
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

setLastWriteTime (Method)
While a file is open for output you may set it’s last write timestamp with this method. The method accepts two parms – a filetimelow and a filetimehigh. Read more about these values here:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/fileio/base/win32_find_data_str.asp

SP2-OCX-DEFINITION:

```
    01  api-setLastWriteTime.
        05  setLastWriteTime-prop-name
                          pic x(30) value "setLastWriteTime".
        05  setLastWriteTime-access-type
                          pic s9(4) comp-5 value +0001.
```

```
05  setLastWriteTime-parm-cnt
              pic s9(4) comp-5 value +0002.
05  setLastWriteTime-value-type.
    10  xString  pic s9(4) comp-5 value +0008.
    10  xEmpty   pic s9(4) comp-5 value +0000.
    10  filler   pic s9(4) comp-5 value +0000.
    10  filler   pic x(4) value low-values.
05  setLastWriteTime-parms.
    10  setLastWriteTime-LowTime pic x(80).
    10  setLastWriteTime-LowTime-type.
       15  xLong    pic s9(4) comp-5 value +0003.
       15  xEmpty   pic s9(4) comp-5 value +0000.
       15  filler   pic s9(4) comp-5 value +0000.
       15  filler   pic x(4) value low-values.
    10  setLastWriteTime-HighTime pic x(80).
    10  setLastWriteTime-HighTime-type.
       15  xLong    pic s9(4) comp-5 value +0003.
       15  xEmpty   pic s9(4) comp-5 value +0000.
       15  filler   pic s9(4) comp-5 value +0000.
       15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move setLastWriteTime-Prop-Name to Sp2-OC-Prop-Name
Move setLastWriteTime-Access-Type to Sp2-OC-Access-Type
Move setLastWriteTime-Parm-Cnt to Sp2-OC-Parm-Cnt
Move setLastWriteTime-Value-Type to Sp2-OC-Value-Type
Move saved-low-time to setLastWriteTime-LowTime
Move saved-high-time to setLastWriteTime-HighTime
Move setLastWriteTime-LowTime to Sp2-OC-Parm-Value (1)
Move setLastWriteTime-LowTime-Type to Sp2-OC-Parm-Type (1)
Move setLastWriteTime-HighTime to Sp2-OC-Parm-Value (2)
Move setLastWriteTime-HighTime-Type to Sp2-OC-Parm-Type (2)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

If the method succeeds “Success” is returned in SP2-OC-VALUE otherwise “Fail” is returned.

apiCloseFile (Method)

This method requires no parameters. It simply closes the file referenced by the internally held file handle. It doesn't matter which of the ApiSimple functions opened the file. There is no return value.

SP2-OCX-DEFINITION:

```
       01  api-apiCloseFile.
           05  apiCloseFile-prop-name
                         pic x(30) value "apiCloseFile".
           05  apiCloseFile-access-type
                         pic s9(4) comp-5 value +0001.
           05  apiCloseFile-parm-cnt
                         pic s9(4) comp-5 value +0000.
           05  apiCloseFile-value-type.
               10  xVoid    pic s9(4) comp-5 value +0024.
               10  xEmpty   pic s9(4) comp-5 value +0000.
               10  filler   pic s9(4) comp-5 value +0000.
               10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiCloseFile-Prop-Name to Sp2-OC-Prop-Name
Move apiCloseFile-Access-Type to Sp2-OC-Access-Type
Move apiCloseFile-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiCloseFile-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

apiCreateDirectory (Method) – Available as of version 1.0.1.4
This method requires one paramenter – the fully qualified directory name to create. The directory name is limited by windows to 248 characters.

SP2-OCX-DEFINITION:

```
       01  api-0000-apiCreateDirctry.
           05  apiCreateDirctry-prop-name
                         pic x(80) value
              "apiCreateDirectory".
           05  apiCreateDirctry-access-type
                         pic s9(4) comp-5 value +0001.
           05  apiCreateDirctry-parm-cnt
                         pic s9(4) comp-5 value +0001.
           05  apiCreateDirctry-value-type.
```

```
          10  xString     pic s9(4) comp-5 value +0008.
          10  xEmpty      pic s9(4) comp-5 value +0000.
          10  filler      pic s9(4) comp-5 value +0000.
          10  filler      pic x(4) value low-values.
       05  apiCreateDirctry-parms.
  * Name Shortened - was:
  * apiCreateDirctry-DirectoryName
          10  apiCreatDrctry-DrctryName pic x(80).
          10  apiCreatDrctry-DrctryName-type.
             15  xString    pic s9(4) comp-5 value +0008.
             15  xEmpty     pic s9(4) comp-5 value +0000.
             15  filler     pic s9(4) comp-5 value +0000.
             15  filler     pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move 1 to Sp2-OC-ID
Move apiCreateDirctry-Prop-Name to Sp2-OC-Prop-Name
Move apiCreateDirctry-Access-Type to Sp2-OC-Access-Type
Move apiCreateDirctry-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiCreateDirctry-Value-Type to Sp2-OC-Value-Type
Move Directory-to-create to sp2-oc-parm-1-l
move 255 to SP2-OC-PARM-LEN-1-L
Move apiCreatDrctry-DrctryName-type to Sp2-OC-Parm-Type (1)
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

findFile (Method) – Available as of version 1.0.1.3
This method requires two paramenters. The Folder name to be searched, and the mask to search for in that folder. It returns a count of the number of matching items found. The retFileName method may be used to return filenames from this list.

SP2-OCX-DEFINITION:

```
 01  api-findFile.
    05  findFile-prop-name
                   pic x(80) value
       "findFile".
    05  findFile-access-type
                   pic s9(4) comp-5 value +0001.
    05  findFile-parm-cnt
                   pic s9(4) comp-5 value +0002.
```

```
   05  findFile-value-type.
      10  xInteger    pic s9(4) comp-5 value +0002.
      10  xEmpty      pic s9(4) comp-5 value +0000.
      10  filler      pic s9(4) comp-5 value +0000.
      10  filler      pic x(4) value low-values.
   05  findFile-parms.
      10  findFile-folder pic x(80).
      10  findFile-folder-type.
         15  xString    pic s9(4) comp-5 value +0008.
         15  xEmpty     pic s9(4) comp-5 value +0000.
         15  filler     pic s9(4) comp-5 value +0000.
         15  filler     pic x(4) value low-values.
      10  findFile-mask pic x(80).
      10  findFile-mask-type.
         15  xString    pic s9(4) comp-5 value +0008.
         15  xEmpty     pic s9(4) comp-5 value +0000.
         15  filler     pic s9(4) comp-5 value +0000.
         15  filler     pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move 1 to Sp2-OC-ID
Move findFile-Prop-Name to Sp2-OC-Prop-Name
Move findFile-Access-Type to Sp2-OC-Access-Type
Move findFile-Parm-Cnt to Sp2-OC-Parm-Cnt
Move findFile-Value-Type to Sp2-OC-Value-Type
Move find-folder to Sp2-OC-Parm-Value (1)
Move findFile-folder-type to Sp2-OC-Parm-Type (1)
Move find-mask to Sp2-OC-Parm-Value (2)
Move findFile-mask-Type to Sp2-OC-Parm-Type (2)
Move 2 to sp2-oc-options
CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

retFileName (Method) – Available as of version 1.0.1.3
This method requires one paramenter. The single parameter is the occurance in the table of names returned from the findFile method invocation.

SP2-OCX-DEFINITION:

```
01  api-retFileName.
   05  retFileName-prop-name
                 pic x(80) value
      "retFileName".
```

```
        05  retFileName-access-type
                        pic s9(4) comp-5 value +0001.
        05  retFileName-parm-cnt
                        pic s9(4) comp-5 value +0001.
        05  retFileName-value-type.
            10  xString     pic s9(4) comp-5 value +0008.
            10  xEmpty      pic s9(4) comp-5 value +0000.
            10  filler      pic s9(4) comp-5 value +0000.
            10  filler      pic x(4) value low-values.
        05  retFileName-parms.
            10  retFileName-element pic x(80).
            10  retFileName-element-type.
                15  xInteger   pic s9(4) comp-5 value +0002.
                15  xEmpty     pic s9(4) comp-5 value +0000.
                15  filler     pic s9(4) comp-5 value +0000.
                15  filler     pic x(4) value low-values.
```

Example:

```
                Move Low-Values To Sp2-OC-Data
                Move 1 to Sp2-OC-ID
                Move retFileName-Prop-Name to Sp2-OC-Prop-Name
                Move retFileName-Access-Type to Sp2-OC-Access-Type
                Move retFileName-Parm-Cnt to Sp2-OC-Parm-Cnt
                Move retFileName-Value-Type to Sp2-OC-Value-Type
                Move file-occurance to Sp2-OC-Parm-Value (1)
                Move retFileName-element-Type to Sp2-OC-Parm-Type (1)
                Move 2 to sp2-oc-options
                CALL "SP2" using sp2-set-ocx sp2-ocx-parm
```

ASYNC Communications Functions

ApiSimple supports some basic asynchronous (Serial) communications via the Windows API. Some familiarization with async communications is assumed on the part of the user. Microsoft has some good documentation on serial communications using the Windows API here:

http://msdn.microsoft.com/library/en-us/dnfiles/html/msdn_serial.asp

ApiSimple simplified much of this for you. Three types of flow control are offered. The default flow control, if no other is specified is XON/XOFF. Optionally, you may set the hardwraeFlowControl property to true for hardware (typically called RTS/CTS flow control) or you may specify no flow control by setting the noFlowControl property to true. These two properties are BOOLEAN and set to true by setting the property to a value of 1. These properties may be set anytime before an apiOpenComPort method is used. The routines are even in use from Unix servers doing serial communications via windows clients under Flexus Thin Client.

hardwareFlowControl (Property)
Setting this property true – or to a value of 1 – enables hardware flow control instead of using the default Xon/Xoff. It must be set before the apiOpenComPort method is used.

SP2-OCX-DEFINITION:

```
    01  api-hardwareFlowControl.
        05  hardwareFlowControl-prop-name
                      pic x(30) value "hardwareFlowControl".
        05  hardwareFlowControl-access-type
                      pic s9(4) comp-5 value +0000.
        05  hardwareFlowControl-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  hardwareFlowControl-value-type.
            10  xBoolean pic s9(4) comp-5 value +0011.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move hardwareFlowControl-Prop-Name to Sp2-OC-Prop-Name
Move hardwareFlowControl-Access-Type to Sp2-OC-Access-Type
Move hardwareFlowControl-parm-Cnt to Sp2-OC-Parm-Cnt
Move hardwareFlowControl-Value-Type to Sp2-OC-Value-Type
Move “1” to sp2-oc-value
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

noFlowControl (Property)
Setting this property true – or to a value of 1 – disables flow control instead of using the default Xon/Xoff flow control. It must be set before the apiOpenComPort method is used.

SP2-OCX-DEFINITION:

```
    01  api-noFlowControl.
    05  noFlowControl-prop-name
                 pic x(30) value "noFlowControl".
    05  noFlowControl-access-type
                 pic s9(4) comp-5 value +0000.
    05  noFlowControl-parm-cnt
                 pic s9(4) comp-5 value +0000.
    05  noFlowControl-value-type.
        10  xBoolean pic s9(4) comp-5 value +0011.
        10  xEmpty   pic s9(4) comp-5 value +0000.
        10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move noFlowControl-Prop-Name to Sp2-OC-Prop-Name
Move noFlowControl-Access-Type to Sp2-OC-Access-Type
Move noFlowControl-parm-Cnt to Sp2-OC-Parm-Cnt
Move noFlowControl-Value-Type to Sp2-OC-Value-Type
Move “1” to sp2-oc-value
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

Serial communications works much the same as bytestream file I/O. Com ports are open as files and have a file handle that ApiSimple tracks internally. Only one file – binary or com port may be open at any one time using a single instance of ApiSimple. ApiSimple opens the com ports for I/O allowing data to flow in both directions. To start using serial communications, first execute an apiOpenComPort method.

apiOpenComPort (Method)
This method opens the com port specified with the settings specified as parameters to the call. It accepts two parms. First – the com port. This is something known to windows, such as COM1, COM2 or even COM5 if you have such a com port installed. The second parameter is the device control string setting as specified in the Windows documentation notes above. Something like 9600,n,8,1 is a valid device control string for 9600 baud, no parity, 8 data bits and one stop bit.

SP2-OCX-DEFINITION:

```
 01  api-apiOpenComPort.
    05  apiOpenComPort-prop-name
                 pic x(30) value "apiOpenComPort".
    05  apiOpenComPort-access-type
                    pic s9(4) comp-5 value +0001.
    05  apiOpenComPort-parm-cnt
                    pic s9(4) comp-5 value +0002.
    05  apiOpenComPort-value-type.
        10  xString  pic s9(4) comp-5 value +0008.
        10  xEmpty   pic s9(4) comp-5 value +0000.
        10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
    05  apiOpenComPort-parms.
        10  apiOpenComPort-thePort pic x(80).
        10  apiOpenComPort-thePort-type.
            15  xString  pic s9(4) comp-5 value +0008.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
        10  apiOpenComPort-DeviceControlString pic x(80).
        10  apiOpenComPort-DeviceControlString-type.
            15  xString  pic s9(4) comp-5 value +0008.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiOpenComPort-Prop-Name to Sp2-OC-Prop-Name
Move apiOpenComPort-Access-Type to Sp2-OC-Access-Type
Move apiOpenComPort-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiOpenComPort-Value-Type to Sp2-OC-Value-Type
Move “COM1” to apiOpenComPort-thePort
MOVE “9600,n,8,1” to apiOpenComPort-DeviceControlString
Move apiOpenComPort-thePort to Sp2-OC-Parm-Value (1)
Move apiOpenComPort-thePort-Type to Sp2-OC-Parm-Type (1)
Move apiOpenComPort-DeviceControlString to Sp2-OC-Parm-Value (2)
Move apiOpenComPort-DeviceControlString-Type to Sp2-OC-Parm-Type (2)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The return value in SP2-OC-VALUE from the method call is “Success” if the com port is open and initialized successfully. If it fails, the return value can be one of the following:

“Failed z” – usually means the com port is in use by another application or does not exist. A file handle to the com port could not be obtained.

“Failed 0” – the timeout values for the port could not be set. By default ApiSimple use a 2 second timeout on reads. No timeout is specified for writes.

“Fail Build” – Com ports are controlled by a data control block. If ApiSimple can’t build this block, this value is returned.

“Fail Set” – returned when there is a failure setting the Communications State.

“Fail Clear” – returned when there is a failure clearing the communications status flags.

apiCloseComPort (Method)
This method closes the open com port. It returns “Success” SP2-OC-VALUE if the com port is closed, and “Fail” in the event it cannot close the com port.

SP2-OCX-DEFINITION:

```
    01  api-apiCloseComPort.
        05  apiCloseComPort-prop-name
                      pic x(30) value "apiCloseComPort".
        05  apiCloseComPort-access-type
                      pic s9(4) comp-5 value +0001.
        05  apiCloseComPort-parm-cnt
                      pic s9(4) comp-5 value +0000.
        05  apiCloseComPort-value-type.
            10  xString  pic s9(4) comp-5 value +0008.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiCloseComPort-Prop-Name to Sp2-OC-Prop-Name
Move apiCloseComPort-Access-Type to Sp2-OC-Access-Type
Move apiCloseComPort-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiCloseComPort-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
```

CALL “SP2” using sp2-set-ocx sp2-ocx-parm

SP2-OC-VALUE will contain either “Success” or “Fail” after this call.

apiWriteFile (Method)

This method accepts two parameters – the data buffer to write and that buffers length. As mentioned in the bytestream I/O if you need to write trailing spaces or trailing low-values, append your data buffer with a character – such as * on byte past the end of the true data. Set the second parm, the length to the actual length of the buffer. The example will show sending three bytes “AB “ using this method. For faster thin client performance do not set sp2-oc-options to 2 and do not check for the normal return value of the number of bytes written in Sp2-OC-VALUE.

NOTE: This method can also be used with Bytestream I/O file writes.

SP2-OCX-DEFINITION:

```
01  api-apiWriteFile.
    05  apiWriteFile-prop-name
                   pic x(30) value "apiWriteFile".
    05  apiWriteFile-access-type
                   pic s9(4) comp-5 value +0001.
    05  apiWriteFile-parm-cnt
                   pic s9(4) comp-5 value +0002.
    05  apiWriteFile-value-type.
        10  xLong    pic s9(4) comp-5 value +0003.
        10  xEmpty   pic s9(4) comp-5 value +0000.
        10  filler   pic s9(4) comp-5 value +0000.
        10  filler   pic x(4) value low-values.
    05  apiWriteFile-parms.
        10  apiWriteFile-buffer pic x(80).
        10  apiWriteFile-buffer-type.
            15  xString  pic s9(4) comp-5 value +0008.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
        10  apiWriteFile-bytes pic x(80).
        10  apiWriteFile-bytes-type.
            15  xLong    pic s9(4) comp-5 value +0003.
            15  xEmpty   pic s9(4) comp-5 value +0000.
            15  filler   pic s9(4) comp-5 value +0000.
            15  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move apiWriteFile-Prop-Name to Sp2-OC-Prop-Name
Move apiWriteFile-Access-Type to Sp2-OC-Access-Type
Move apiWriteFile-Parm-Cnt to Sp2-OC-Parm-Cnt
Move apiWriteFile-Value-Type to Sp2-OC-Value-Type
Move “AB *” to apiWriteFile-buffer
MOVE “3” to apiWriteFile-bytes
Move apiWriteFile-buffer to Sp2-OC-Parm-Value (1)
Move apiWriteFile-buffer-Type to Sp2-OC-Parm-Type (1)
Move apiWriteFile-bytes to Sp2-OC-Parm-Value (2)
Move apiWriteFile-bytes to Sp2-OC-Parm-Type (2)
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

The number of bytes written will be returned in SP2-OC-VALUE

readData (Method)
This method reads data from the COM port and stores it in the myData property for retrieval. Unlike the apiWriteFile method, this method is only valid for serial communications. Before calling this method you ***MUST*** set the maxBytes property to a value < 8192 specifying the maximum number of bytes you expect to read. NOTE: A smaller number may actually be returned. Do not count on the read actually returning the number specified. Check Sp2-OC-VALUE after a call to the readData method to determine the number of bytes read.

maxBytes (Property)
Set this property to the maximum number of bytes you desire to have returned from a readData method execution. This value may not exceed 8192.

SP2-OCX-DEFINITION:

```
    01  api-maxBytes.
        05  maxBytes-prop-name
                     pic x(30) value "maxBytes".
        05  maxBytes-access-type
                     pic s9(4) comp-5 value +0000.
        05  maxBytes-parm-cnt
                     pic s9(4) comp-5 value +0000.
        05  maxBytes-value-type.
```

```
            10  xUnsShortpic s9(4) comp-5 value +0018.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move maxBytes-Prop-Name to Sp2-OC-Prop-Name
Move maxBytes-Access-Type to Sp2-OC-Access-Type
Move maxBytes-parm-Cnt to Sp2-OC-Parm-Cnt
Move maxBytes-Value-Type to Sp2-OC-Value-Type
Move “80” to sp2-oc-value
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

readData (Method) Definition and Example

SP2-OCX-DEFINITION:

```
     01  api-readData.
        05  readData-prop-name
                     pic x(30) value "readData".
        05  readData-access-type
                     pic s9(4) comp-5 value +0001.
        05  readData-parm-cnt
                     pic s9(4) comp-5 value +0000.
        05  readData-value-type.
            10  xInteger pic s9(4) comp-5 value +0002.
            10  xEmpty   pic s9(4) comp-5 value +0000.
            10  filler   pic s9(4) comp-5 value +0000.
            10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move readData-Prop-Name to Sp2-OC-Prop-Name
Move readData-Access-Type to Sp2-OC-Access-Type
Move readData-Parm-Cnt to Sp2-OC-Parm-Cnt
Move readData-Value-Type to Sp2-OC-Value-Type
*> Next Line Required if used with Thin Client – will work without thin client
Move 2 to sp2-oc-options
CALL “SP2” using sp2-set-ocx sp2-ocx-parm
```

SP2-OC-VALUE will contain the number of bytes read. You will need to convert this text value to a number, perhaps using the NUMVAL intrinsic function.

To access the data read, retrieve the value of the myData property.

myData (property)
This property contains data read from a file or com port. It's max size is 8192. Use the length returned from the read operation to get at the data.
NOTE: You may want to use the USR-OCX-PARMS definition included in USROCX.CPY since the parms are defined to be a larger size than the default 80 bytes of sp2ocx.cpy.

SP2-OCX-DEFINITION:

```
   01  api-myData.
       05  myData-prop-name
                     pic x(30) value "myData".
       05  myData-access-type
                     pic s9(4) comp-5 value +0000.
       05  myData-parm-cnt
                     pic s9(4) comp-5 value +0000.
       05  myData-value-type.
           10  xString  pic s9(4) comp-5 value +0008.
           10  xEmpty   pic s9(4) comp-5 value +0000.
           10  filler   pic s9(4) comp-5 value +0000.
           10  filler   pic x(4) value low-values.
```

Example:

```
Move Low-Values To Sp2-OC-Data
Move ocx-field-id to Sp2-OC-ID
Move myData-Prop-Name to Sp2-OC-Prop-Name
Move myData-Access-Type to Sp2-OC-Access-Type
Move myData-Parm-Cnt to Sp2-OC-Parm-Cnt
Move myData-Value-Type to Sp2-OC-Value-Type
CALL “SP2” using sp2-get-ocx sp2-ocx-parm
```

The result will be returned in SP2-OC-VALUE

Performance Considerations

Since the control is fairly large, it does take some time to load. If used frequently you will wish to avoid this. The sample – APISHELL.CBL shows how to load the control in a hidden window – using panelless operation so that it doesn't take up a panel file handle – and how to make repeated calls to it for a subset of it's functions. You can use this program as a template to add additional functions. The sample – TESTCALL.CBL illustrates calling the APISHELL sample program.

Future Enhancements

As a licensed ApiSimple user, your input is valued. If you find a capability, feature or enhancement that you need and you think is a good fit for ApiSimple, please feel free to email your idea to thane@flexus.com for consideration as an addition to the control.