

Using the APLNext.FileWatcher

Overview:

The APLNext.FileWatcher will report change, create, delete and rename file events on a programmer-defined watched folder on the target machine. This capability is often useful when an APL+Win application will perform processing triggered by file changes to a watched folder.

The APLNext.FileWatcher is a .Net 4.0 assembly which exposes its methods, properties and events as ActiveX so that they can be used by Win32 applications such as APL+Win. The APLNext.FileWatcher may also be used as a fully-managed .Net 4.0 assembly in any .Net programming language such as C#. Since the APLNext.FileWatcher is a .Net-based application it can employ multi-threading to provide the file activity information about the watched folder with minimal impact on other running applications.

Installation:

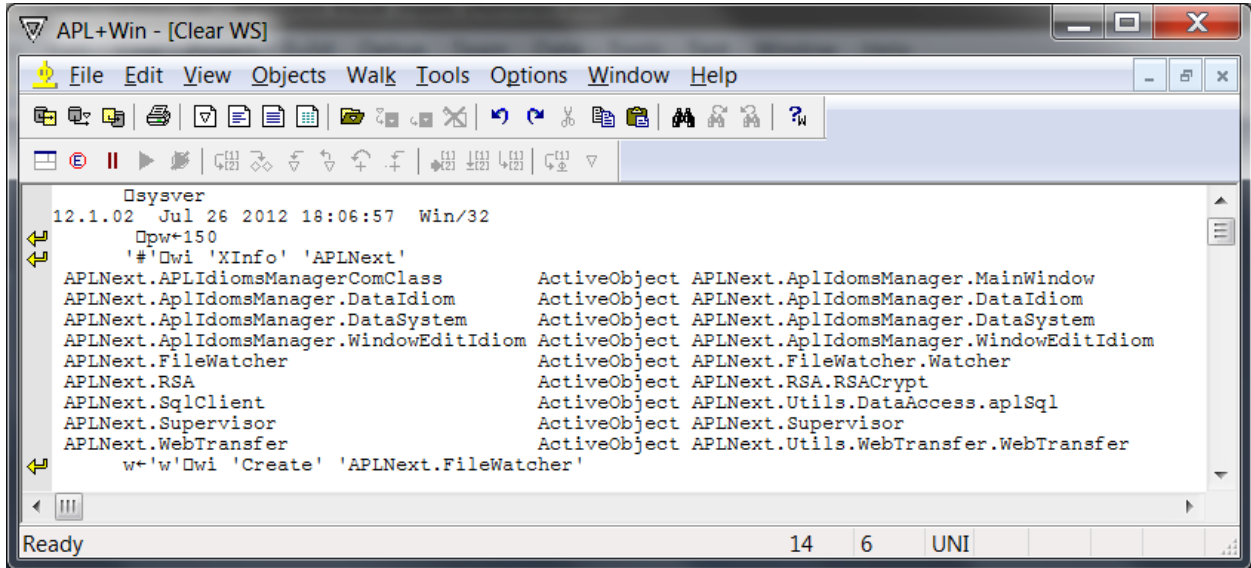
The APLNext.FileWatcher is installed to the target workstation or server using the provided "APLNext.FileWatcherSetup.msi" Microsoft installer. This installer can be used with the developer or run-time version of APL+Win. So that APL+Win can use the APLNext.FileWatcher as an ActiveX component, the APLNext.FileWatcherSetup.msi will register the APLNext.FileWatcher. To perform this registration the APLNext.FileWatcherSetup.msi must be run with Administrator credentials so that it can update the target machine's registry.

Prerequisites:

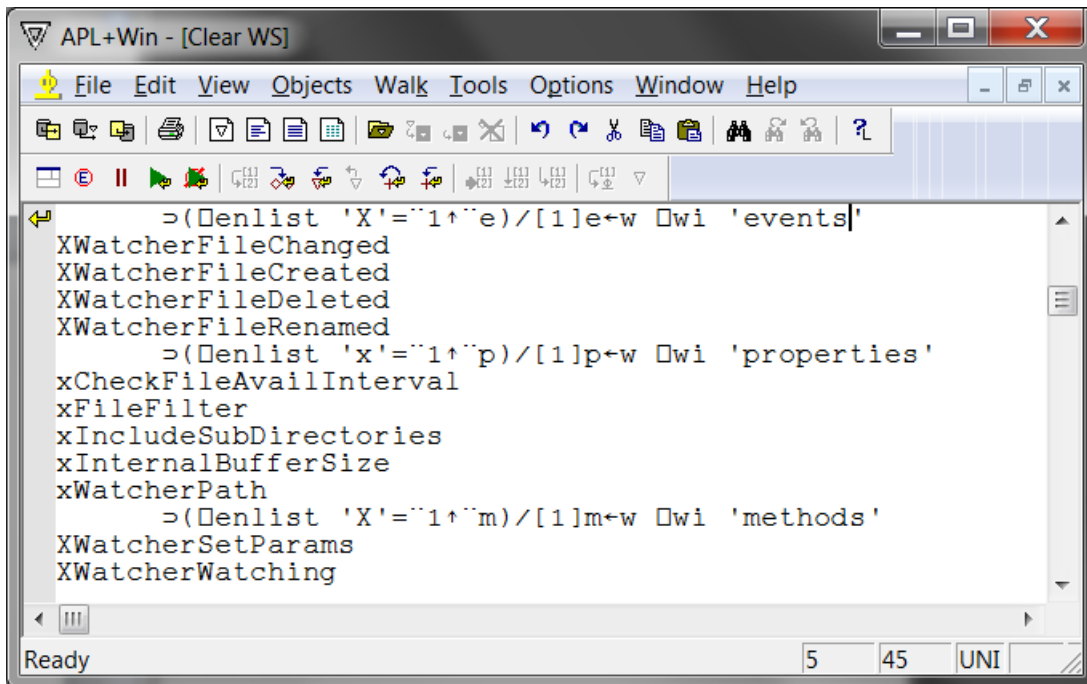
The APLNext.FileWatcher requires the Microsoft .Net 4.0 Framework on the target machine. An instance of the APLNext.FileWatcher can be created by APL+Win v12.0.0 or subsequent version.

Using the APLNext.FileWatcher in APL+Win:

After installing the APLNext.FileWatcher, its 'Progid' will be visible in the 'XInfo' property of the APL+Win system object. An instance of the APLNext.FileWatcher may be created in the APL+Win session.



The APLNext.FileWatcher has an object model with methods, properties and events:



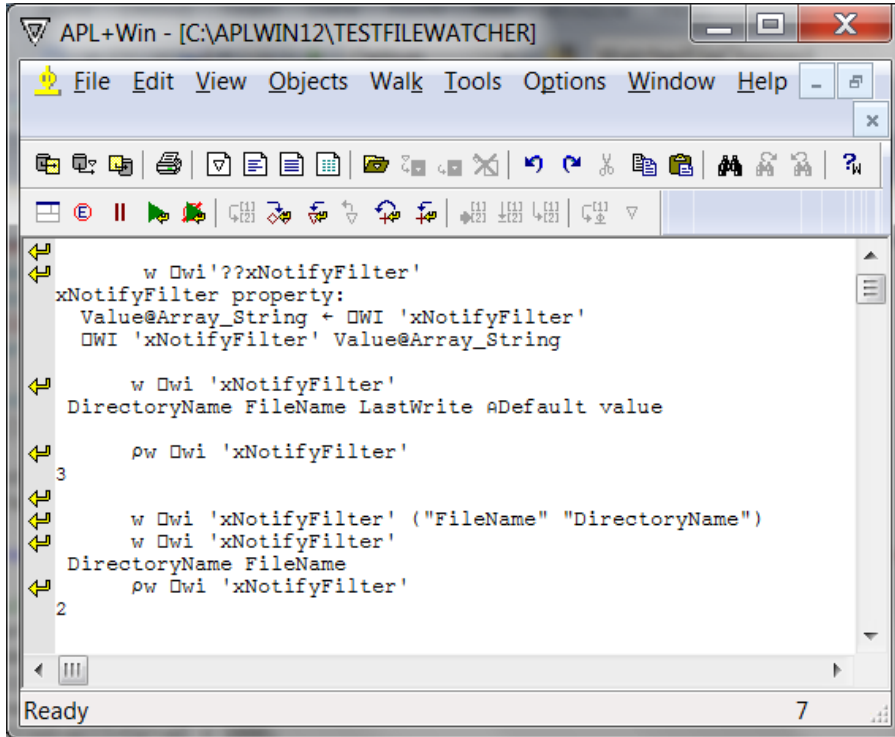
APLNext.FileWatcher Properties:

Use these properties to subsequently set APLNext.FileWatcher property values. The APLNext.FileWatcher 'WatcherSetParams' method may be used to set most of these properties in one operation.

- The '**WatcherPath**' property indicates the path on the target machine that will be watched. For example "c:\my watched folder". The default value is "". If the 'WatcherPath' path does not exist the APLNext.FileWatcher will throw an exception. Note that changes in folders that contain the watched folder will not be captured by the APLNext.FileWatcher.
- The '**IncludesSubDirectories**' property indicates if subdirectories of the watched folder will be included in the watch. For example 1/'True'. The default value is 0/'False'.
- The '**FileFilter**' property indicates which file types will be watched, e.g. *.*'/All files, *.txt'/Text files. The default value is *.*'.
- The '**InternalBufferSize**' is the size in bytes of the buffer used to record the events being watched. It must be in the range [4096, 65536]. The default value is 4096. Increase the 'InternalBufferSize' when many watched file operations, each of which takes significant time to process, are expected by the application system.
- The '**CheckFileAvailInterval**' is the timespan in milliseconds between the file operation occurring and the event notification being sent by the APLNext.FileWatcher. The default value is 2000 (2 seconds). This time interval should be no less than the length of time to complete a watched file operation anticipated by the application system. Setting this time interval too low may result file permission conflicts because the APL+Win application system is attempting to access a watched file before the operating system file system operation has been completed.
- The '**NotifyFilter**' is the collection of file system information items which will be checked to determine if a file system operation has occurred and will thus trigger an event notification. The value for this property is a one-dimensional array of strings selected from the list below. These strings are case-sensitive. For example ("Attributes" "LastWrite"). The default value is ("FileName" "DirectoryName" "LastWrite").
 - Attributes
 - DirectoryName
 - FileName
 - LastAccess
 - LastWrite
 - Security
 - Size

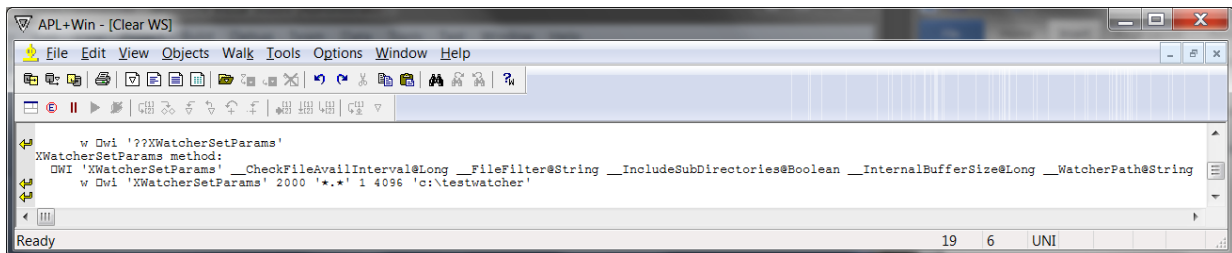
Setting the 'NotifyFilter' Property:

This property is set as a one-dimensional array of text strings with each element corresponding to an included NotifyFilter condition.



APLNext.FileWatcher 'WatcherSetParams' Method:

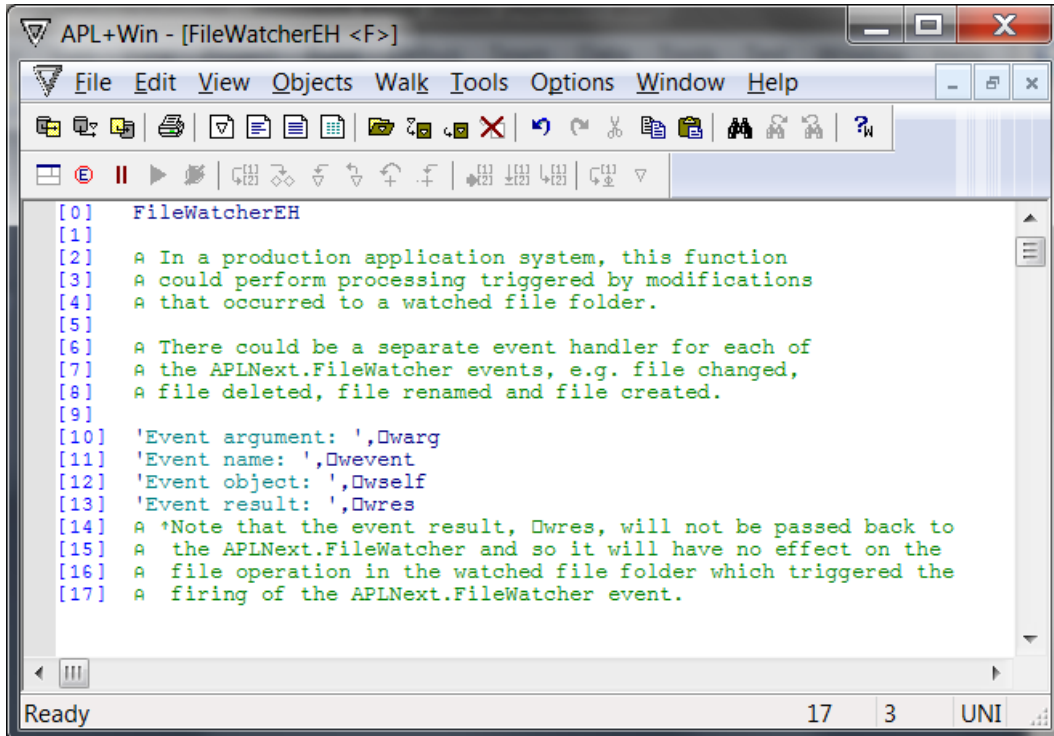
This method is convenient to set the APLNext.FileWatcher parameters, except for the 'NotifyFilter' property, in a single operation.



APLNext.FileWatcher 'WatcherFileCreated' Event:

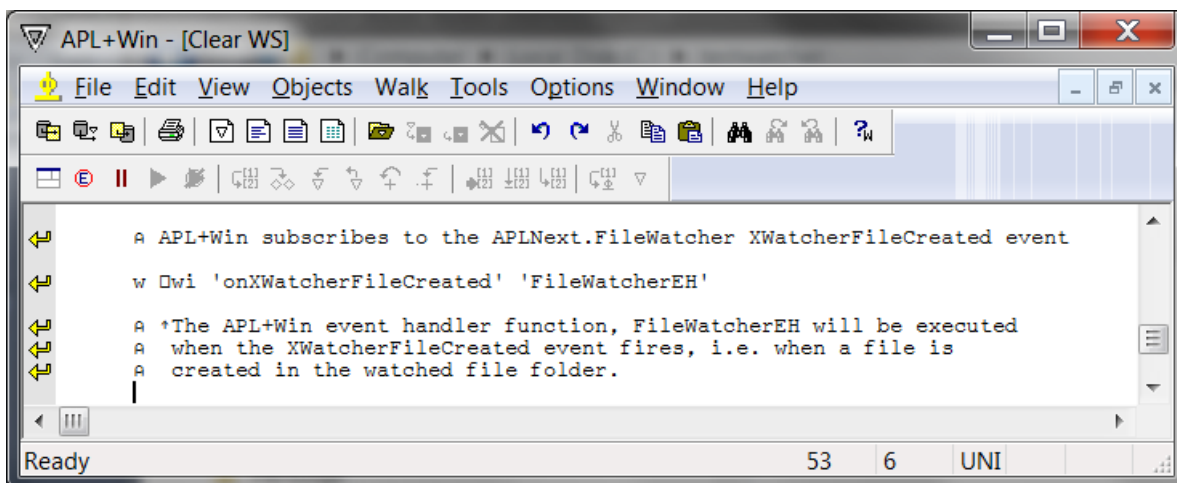
If APL+Win will subscribe to the 'WatcherFileCreated' event, define the APL+Win event handler function for this event and then subscribe to the 'WatcherFileCreated' event. For the 'WatcherFileCreated' event the value of []warg is the full path of the file that was created in the watched folder.

A sample APL+Win event handler function:



```
APL+Win - [FileWatcherEH <F>]
File Edit View Objects Walk Tools Options Window Help
[0] FileWatcherEH
[1]
[2] A In a production application system, this function
[3] A could perform processing triggered by modifications
[4] A that occurred to a watched file folder.
[5]
[6] A There could be a separate event handler for each of
[7] A the APLNext.FileWatcher events, e.g. file changed,
[8] A file deleted, file renamed and file created.
[9]
[10] 'Event argument: ',Dwarg
[11] 'Event name: ',Dwevent
[12] 'Event object: ',Dwself
[13] 'Event result: ',Dwres
[14] A *Note that the event result, Dwres, will not be passed back to
[15] A the APLNext.FileWatcher and so it will have no effect on the
[16] A file operation in the watched file folder which triggered the
[17] A firing of the APLNext.FileWatcher event.
Ready 17 3 UNI
```

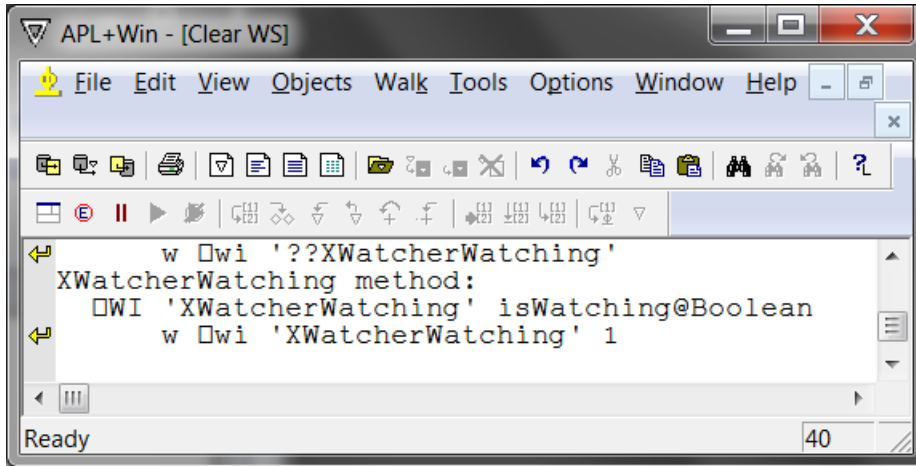
APL+Win subscribes to the 'XWatcherFileCreated' event of the APLNext.FileWatcher:



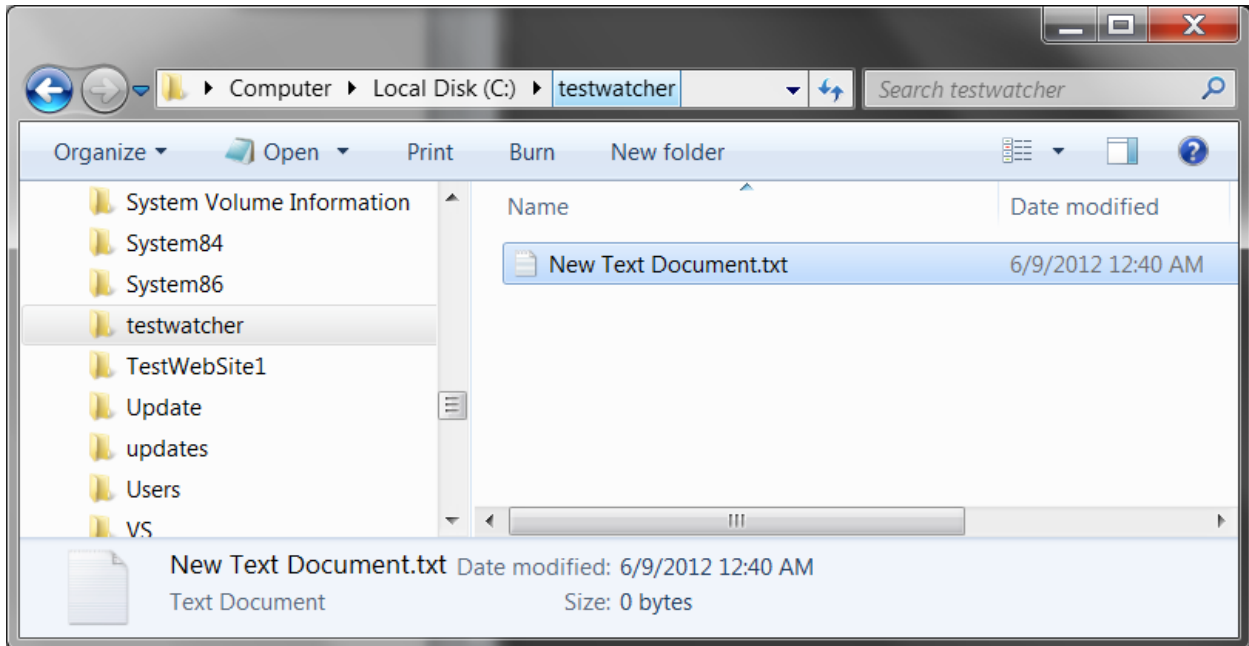
```
APL+Win - [Clear WS]
File Edit View Objects Walk Tools Options Window Help
A APL+Win subscribes to the APLNext.FileWatcher XWatcherFileCreated event
w Dwi 'onXWatcherFileCreated' 'FileWatcherEH'
A *The APL+Win event handler function, FileWatcherEH will be executed
A when the XWatcherFileCreated event fires, i.e. when a file is
A created in the watched file folder.
Ready 53 6 UNI
```

APLNext.FileWatcher 'WatcherWatching' Method:

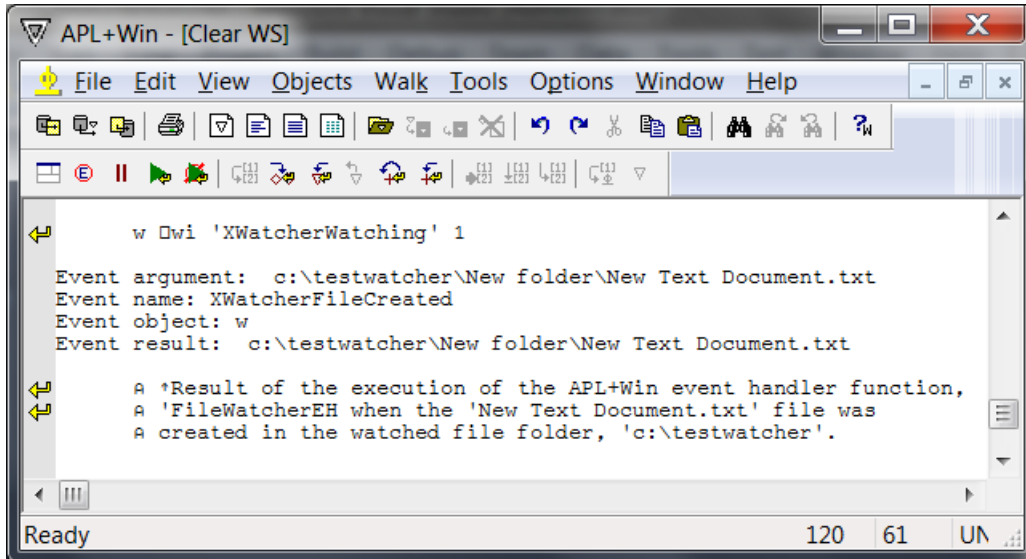
Before using this method to commence watching the selected folder, define the APL+Win event handlers which will be used to subscribe to the APLNext.FileWatcher events. Start (or stop) the APLNext.FileWatcher watching the watched folder using this APLNext.FileWatcher method and the argument 1/'True' (or 0/'False') respectively



To see the APLNext.FileWatcher in action from APL+Win, create a file in, or copy a file into, the watched folder:



In this example APL+Win 'FileCreatedEH' event handler will display the full path of the new file that was created in the watched folder. In a production application system the APL+Win event handler will perform the application-specific processing triggered by the subscribed events occurring in the watched folder. Note that an operating system file operation which appears to the user to be a single event may generate multiple APLNext.FileWatcher event notifications. See the important note below describing this behavior in more detail.



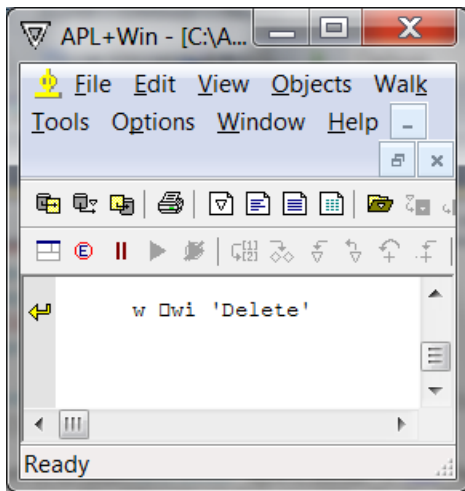
Other APLNext.FileWatcher Events:

The APLNext.FileWatcher 'WatcherFileChanged' and 'WatcherFileDeleted' events are analogous to the 'WatcherFileCreated' event since the []warg value is the full path of the file in the watched folder which was changed, deleted or created, respectively.

The APLNext.FileWatcher 'WatcherFileRenamed' event []warg contains two values, the full path of the file in the watched folder after the rename has occurred and the full path of the file which was renamed before it was renamed.

Delete the APLNext.FileWatcher Instance after it is no longer needed:

Use the 'Delete' method to recover system resources:



APL+Win Event Variables:

In APL+Win there are several system variables which are defined by APL+Win for use within an event handler. The []wself, []wevent and []warg variables are available for use when APL+Win subscribes to an event exposed by the APLNext.FileWatcher. The []wres system variable represents a custom messaging methodology unique to APL+Win. The APLNext.FileWatcher is a .Net assembly which implements its own event structure which is distinct from that of APL+Win. The value of []wres cannot be passed back from the APL+Win event handler to .Net. So modifying []wres will have no effect on the file operation in the watched file folder which triggered the firing of the APLNext.FileWatcher event.

IMPORTANT: Multiple Event Notifications for Apparently One File Event:

The underlying technology of the APLNext.FileWatcher is the Microsoft .Net 'System.IO.FileSystemWatcher' class. Some file operations which appear to be a single event, generate multiple APLNext.FileWatcher event notifications. For example when using Microsoft Notepad to create a new file in the watched directory, generally two file creation event notifications will occur, because Notepad performs multiple file system actions, writing the file content and writing the file attributes.

Other file operations, such as copy/paste of a file, may perform multiple operations in the same manner, for example editing a Microsoft Word or Excel file involves creating temporary file copies and updating the content and permissions of the original source files.

Modifying the 'NotifyFilter' property of the APLNext.FileWatcher to select a limited number of indicators of a file operation may reduce the incidence of multiple event notifications.

This behavior of certain watched file operations to trigger multiple APLNext.FileWatcher event notifications means that the application system functions which perform the work triggered by the watched file operations must be designed so that they can recognize when the work they perform has already been completed when multiple events associated with the same file system operation are received.

For example, suppose that the APL+Win application system will read the contents of a new file in the 'c:\WatchedFolder' and put that information in an APL+Win component file data store for subsequent processing. To implement this algorithm:

- The file created file system operation will be watched by the APL+Win application system by subscribing to the APLNext.FileWatcher 'WatcherFileCreated' event in the 'TestFileWatcher' function.
- The 'FileWatcherEH' event handler function will be executed when the 'WatcherFileCreated' event occurs:
 - The file name of the created file in is checked in a files processed list maintained by the APL+Win application system to see if it has already been processed.
 - If the created file has not already been processed, the APL+Win application system:
 - Reads the content of the new file
 - Performs an application-specific check for non-empty content
 - If there is non-empty content:
 - Updates the files processed list
 - Appends the file content to the APL+Win application data store
- The APL+Win function listings below illustrate the implementation of this example algorithm. The sample APL+Win functions are illustrative only and do not represent a full production implementation which would involve considerations such as exception handling and performance.

```

APL+Win - [TestFileWatcher <F>]
File Edit View Objects Walk Tools Options Window Help
[0] TestFileWatcher
[1]
[2] 'AppDataStore.cf' Dofcreate 1
[3] A 'APL+Win application system data store
[4] A Each component contains the new created file content
[5] A
[6] FilesAlreadyProcessed+OpC''
[7] A 'FilesAlreadyProcessed is a vector of filenames
[8] A already processed by the APL+Win application system
[9] A as their creation event is received.
[10]
[11] w+'w'Dwi 'Create' 'APLNext.FileWatcher'
[12] w Dwi 'XWatcherSetParams' (20*1000) '*.*' 1 (8*1024) 'c:\WatchedFolder'
[13] A 'Delay 20 sec. for all WatcherFileCreated events to fire
[14] w Dwi 'onXWatcherFileCreated' 'FileWatcherEH'
[15] w Dwi 'XWatcherWatching' 1
Ready 0 0 UNI

```

```

APL+Win - [FileWatcherEH <F>]
File Edit View Objects Walk Tools Options Window Help
[0] FileWatcherEH;CONTENT
[1] :IF ~(C=Dwarg)∈FilesAlreadyProcessed
[2]   Dnuntie -1
[3]   (Denlist Dwarg) Dxntie -1
[4]   CONTENT+Dnread -1 82 (Dnsize -1) 0
[5]   Dnuntie -1
[6]   :IF ~""=CONTENT
[7]     A 'Application-specific test for empty content
[8]     A to determine if the client application has added content to the new file
[9]     FilesAlreadyProcessed+FilesAlreadyProcessed,C=Dwarg
[10]    CONTENT Dofappend 1
[11]    'File processed: ',Dwarg
[12]   :ELSE
[13]     'New file detected, but no content: ',Dwarg
[14]     A 'This condition may be a symptom of too small a value for the
[15]     A CheckFileAvailInterval property of the APLNext.FileWatcher
[16]   :ENDIF
[17]   :ELSE
[18]     'File already processed: ',Dwarg
[19]   :ENDIF
Ready 0 0 UNI

```

See here for more information on multiple file system operation events:
<http://msdn.microsoft.com/en-us/library/system.io.filesystemwatcher.created.aspx>

Deployment of the APLNext.FileWatcher:

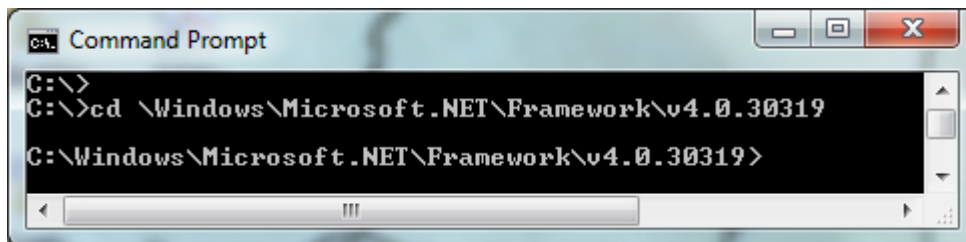
The APLNext.FileWatcher must be registered as an ActiveX component so that APL+Win can utilize it. Administrator credentials will be necessary to install the APLNext.FileWatcher on the target workstation and register it as an ActiveX component accessible to APL+Win.

Use the included Windows Installer: APLNext.FileWatcherSetup.msi:

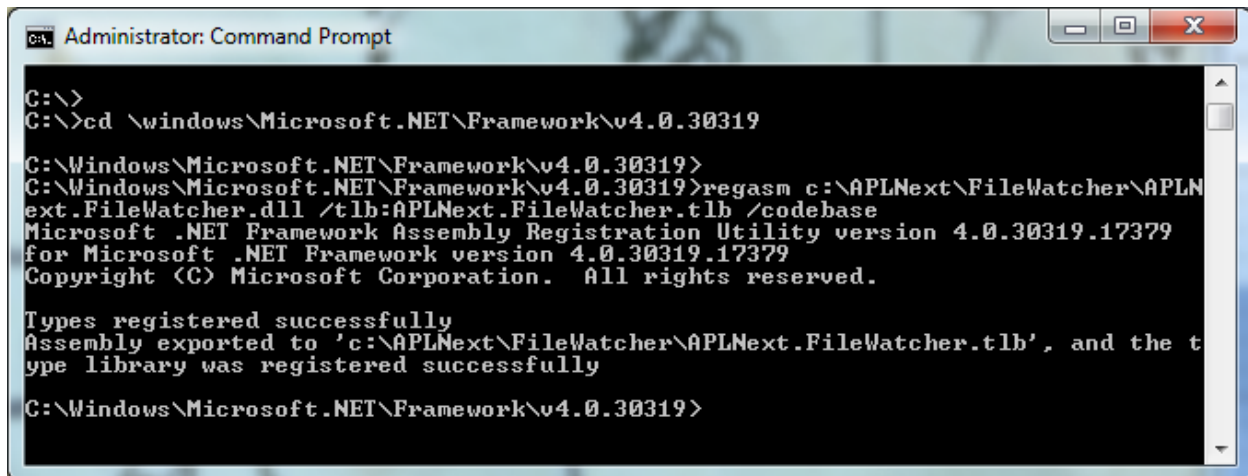
The easiest option for deployment of the APLNext.FileWatcher is to run the "APLNext.FileWatcherSetup.msi" Microsoft installer on the target workstation. This installer can be concatenated onto a larger application system installer if desired. Using the "APLNext.FileWatcherSetup.msi" Microsoft installer means that removing or repairing the installation is easily done from the Windows Control Panel > Programs & Features dialog.

Manually install and register the APLNext.FileWatcher using Microsoft regasm.exe:

- Create a subdirectory on the target workstation containing the "APLNext.FileWatcher.dll" and the "APLNext.TestForAPLWin.dll" files, for example "c:\APLNext\FileWatcher\". This will be the permanent location on this workstation of the required components of the APLNext.FileWatcher utility. To obtain copies of these files, run the "APLNext.FileWatcherSetup.msi" on a different workstation.
- Use Programs > Accessories and right click the Command Prompt menu item and select "Run As Administrator".
- Use the command "cd \windows\microsoft.net\framework\v4.0.30319" to change to the directory containing the appropriate version of the "regasm.exe" program. Since the APLNext.FileWatcher is based on the Microsoft .Net Framework 4.0, the appropriate "regasm.exe" program is the .Net 4.0 version.



- Use the command "regasm c:\APLNext\FileWatcher\APLNext.FileWatcher.dll /tlb:APLNext.FileWatcher.tlb /codebase" to register the APLNext.FileWatcher as an ActiveX component on the target machine. The "/tlb" switch will create the "c:\APLNext\FileWatcher\APLNext.FileWatcher.tlb" file that is used by the APL+Win ActiveX interface to discover and use the methods, properties and events associated with the APLNext.FileWatcher. The "/codebase" switch is used because the APLNext.FileWatcher dlls are not located in the Microsoft .Net 'global assembly cache' of the target workstation.

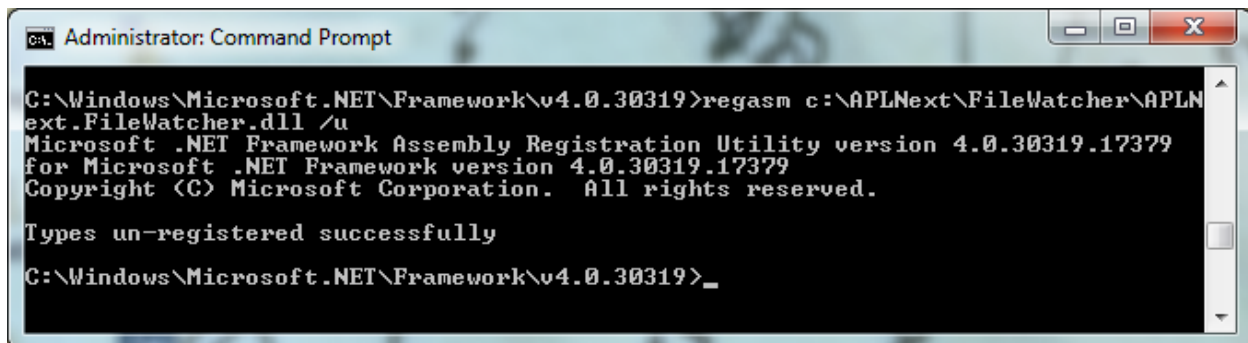


```
Administrator: Command Prompt
C:\>
C:\>cd \windows\Microsoft.NET\Framework\v4.0.30319
C:\Windows\Microsoft.NET\Framework\v4.0.30319>
C:\Windows\Microsoft.NET\Framework\v4.0.30319>regasm c:\APLNext\FileWatcher\APLNext.FileWatcher.dll /tlb:APLNext.FileWatcher.tlb /codebase
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.17379
for Microsoft .NET Framework version 4.0.30319.17379
Copyright (C) Microsoft Corporation. All rights reserved.

Types registered successfully
Assembly exported to 'c:\APLNext\FileWatcher\APLNext.FileWatcher.tlb', and the type library was registered successfully

C:\Windows\Microsoft.NET\Framework\v4.0.30319>
```

If the APLNext.FileWatcher is manually registered, the “regasm.exe” program is used to un-register it should it be necessary. To manually un-register the APLNext.FileWatcher, use “regasm.exe” with the “/u” parameter. When using the “regasm.exe” program it is important to note the location of the .Net assembly being registered so that that address can be used to successfully unregister the .Net assembly if desired.



```
Administrator: Command Prompt
C:\Windows\Microsoft.NET\Framework\v4.0.30319>regasm c:\APLNext\FileWatcher\APLNext.FileWatcher.dll /u
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.17379
for Microsoft .NET Framework version 4.0.30319.17379
Copyright (C) Microsoft Corporation. All rights reserved.

Types un-registered successfully

C:\Windows\Microsoft.NET\Framework\v4.0.30319>_
```

For more information on the “regasm.exe” program see: [http://msdn.microsoft.com/en-us/library/tzat5yw6\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/tzat5yw6(v=vs.71).aspx).