

Wired For Sound

Wired for Sound lets you specify which multimedia effects--sounds, videos, or pictures---to play when any of a number of Windows events happen. For instance, you can have any media element you choose play when dialog boxes are displayed, applications are launched, or system errors occur.

Attaching Sounds, Videos, and Pictures to Windows Events

Wired for Sound Buttons

Association Files

Creating a New Association File

Opening an Existing Association File

Saving the Current Associations

Setting the Default Association File

Button Events

Creating Button Events

Editing Button Events

Keyboard Events

System Error Events

Startup Events

Icon Events

Creating Icon Events

Editing Icon Events

Window Events

Creating Window Events

Editing Window Events

Preferences

Attaching Sounds, Videos, and Pictures to Windows Events

When you attach one or more media elements to a Windows event, you specify which media elements, if any, will be played when the event occurs.

1. Determine which type of event you want to attach media elements to. Wired for Sound lets you attach media elements to the following types of events:

Button events

System Error events

Icon events

Startup events

Keyboard events

Window events

2. Either click on the appropriate Event button or choose the appropriate item from the Edit menu.
3. Choose the specific event you want to attach media elements to from the Events list. The media elements that are currently associated with this event will be listed in the Media Elements list.
4. Press Add... to add media elements to the list of media elements for the selected event. You can mix and match different types of media elements: pictures, videos, and bitmaps.
5. Select the desired playback options for the media elements. You can choose to have either one or all elements play each time the event happens. You can also specify whether the events play in sequential (listed) or random order.
6. If the media elements will be playing in sequential order, you can use the up, down, and sort buttons to arrange the list to the desired order.
7. Make sure that the Disable events button is not checked. If it is, no sounds for any event in the Events list will be played.
8. You may select more events from the Events list and attach sounds to them. When done, press OK. Or, press Cancel to leave all events attached to whatever media elements they previously were.

Multiple sets of associations can be created and used with Wired for Sound. See Association Files for details.

Note

You can create more Window events, Icon events, and Button events if you'd like to attach a sound to a window, dialog box, Program Manager icon, or button that's not already listed under Events.

Wired for Sound Buttons

The main window of Wired for Sound displays several buttons which can be used to attach media elements to Windows events and launch other Wired for Sound applications.

Note that these are buttons, and not icons. You need only click once on them for them to work!

The top six buttons are Event buttons.

The others launch the appropriate Wired for Sound application, such as Talking Clock, Talking Calendar, or other utilities. If they're already running, they'll be activated.

To conserve screen space, you may want to run Wired for Sound with only the Event buttons showing. To do so, choose Small window from the Preferences dialog.

Note:

Wired for Sound searches for the other applications in the same directory where it's located. The install program puts them here by default. If you move them elsewhere, it may not be able to locate them (although its chances are quite good if they're in your path, Windows directory, or Windows system directory).

Association Files

An association file stores information about which media elements will be played when different Windows events happen and which groups of events are enabled or disabled.

With Wired for Sound, you can maintain a number of association files. By default, Wired for Sound stores all your associations in DEFAULT.WFS. You may want to use different association files if, for instance, multiple people use the same machine, or if you'd like to be able to switch easily between having a lot of loud sounds connected to every aspect of Windows and a more quiet, reserved set of associations.

See: [Creating a New Association File](#), [Opening an Existing Association File](#), [Saving the Current Associations](#), [Setting the Default Association File](#).

Creating a New Association File

If you want to maintain multiple configurations, you may want to create separate association files for each configuration. You can make one association file the default for Wired for Sound. See Setting the Default Association File for details.

1. Choose New from the File menu. If you've made changes to the currently loaded association file, you'll be asked to save it.
2. Initially, no media elements will be attached to any events. Follow the instructions in Attaching Sounds, Videos, and Pictures to Windows Events.
3. When done, save the associations with the Save As... command on the File menu. See Saving Association Files.

Opening an Existing Association File

You can load a new association file into Wired for Sound to change the media elements which play for different Windows events.

1. Choose Open... from the File menu.
2. Choose the file you want to open. If you've made changes to the currently loaded association file, you'll be asked to save it first.
3. Press OK. The new associations will now be in effect.

Saving Association Files

You can choose to save the current association file under its current name or choose a new name. If you have not yet specified a name, you'll be asked to supply one.

To save under its current name, choose the Save command from the File menu.

To save under a new name, choose Save As from the File menu. If you want this association file to be the default whenever Wired for Sound is loaded, check the "Make default association file" box.

Setting the Default Association File

The default association file is the association file which is loaded automatically when Wired for Sound is first run. By default, DEFAULT.WFS is loaded. To change this, open the association file you want to be the default, choose the Save As... option from the File menu, and check the "Make default association file" option.

See: [Opening an Existing Association File](#), [Saving the Current Associations](#).

Preferences

To view the Preferences dialog, choose Preferences... from the Edit menu.

Preference	Effect
Small Window	Checking this option will cause Wired for Sound's main window to show only the <u>Event Buttons</u> .
Hidden Icon	Checking this option will cause Wired for Sound's icon to become hidden when you minimize Wired for Sound. You can restore Wired for Sound by re-launching it.
Enable Speaker Buttons	Checking this option will cause the <u>speaker buttons</u> to appear in the title bars of other Windows applications. Unselect this option to speed up window redrawing or improve compatibility if you have problems.
Preempt Sounds	Checking this option will cause Wired for Sound to stop the currently playing sound if an event causes it to play a new sound before a previous event's sound hasn't completed. With this option unchecked, the new event's sound will not be played in this case.
Show Banner	Uncheck this item to prevent Wired for Sound from displaying the logo banner whenever it loads.
Hot key	You can specify a key or mouse button which will activate Wired for Sound no matter which application you're using. This is useful if you use the Hidden Icon option. To specify the hot key, select the Hot key control using the Tab key or the mouse, then press the key or mouse button you want. You can hold down shift, control, and/or alt if you want. Press Backspace to specify no hot key. You cannot use Esc, Enter, Tab, Shift-Tab, or Backspace as the hot key.
Autoload/Don't load	Pressing Autoload will make Wired for Sound load automatically when Windows starts up. Pressing Don't load will turn off this behavior.

Button Events

Wired for Sound allows you to attach [media elements](#) to virtually any button, including push buttons, check boxes, and radio buttons. Wired for Sound comes with [Button events](#) for many common buttons. You can add more Button events to attach media elements to as well. To do this, you first create a Button event. Then, you attach media elements to the new Button event.

See [Creating Button Events](#), [Editing Button Events](#), and [Attaching Sounds, Videos, and Pictures to Windows Events](#) for more details.

Note

Button events are *not* stored in association files. Thus, when you create a new Button event, all your association files will be able to have sounds attached to it, no matter which association file is currently loaded.

Creating Button Events

1. Choose Button Events from the Edit menu.
2. Click on the Add... button to open the Add Button Event dialog.
3. Enter the text which appears on the button face. Don't worry about the fact that some characters may be underlined; Wired for Sound ignores this underlining. You can also have Wired for Sound determine this text by pressing the Pick Button... button and clicking on the appropriate button. Note that the button must not be disabled (grayed) or in a dialog box behind another window or dialog box when you use Pick Button.
4. Type a description for the new button event in the Description field. This description can be whatever you want, but it must be specified. It will be used when you attach sounds to the event later. Note that if you later decide to change the description, you will have to re-attach sounds to it in your other sound association files. By default, the description is the button's text followed by "Button."
5. Click OK to close the Edit Button Event dialog.
6. You can create or edit more button events. When done, press OK.

Editing Button Events

1. Choose Button Events... from the Edit menu.
2. Choose an event from the list and click on the Edit... button to open the Edit Button Event dialog.

You can change the description and button text for the event as described in [Creating Button Events](#).

Note

If you change the description of the event, the media elements in other association files which were previously attached to it will no longer be attached. You will have to re-attach media elements to the event in all the other sound association files.

System Error Events

Wired for Sound allows you to attach [media elements](#) to several system errors. As they are pre-defined by Wired for Sound, you cannot add or edit System Error events.

Event	Description
Divide By Zero	This error is caused by an application which attempts to divide a number by zero. Most applications will not allow this error to pass to Windows. If Windows does detect this error, it will display an error box requesting that you terminate the offending application.
Invalid Opcode	This error is caused by an application trying to execute a bad processor command. Normally, Windows displays a message box explaining that there was an application error and allowing you to Close the application or Ignore the error.
Protection Fault	This error is the celebrated General Protection Fault (formerly called the UAE). It happens when an application accesses memory it doesn't own, or in a manner it's not allowed to. Generally, it's caused by a programming bug. Normally, Windows displays a message box explaining that there was an application error and allows you to Close the application or Ignore the error.
Stack Fault	This error is similar to the UAE, and is handled similarly. It is quite rare, even in buggy applications!

Keyboard and Mouse Events

Wired for Sound allows you to attach media elements to many of the keys on your keyboard, and to various mouse-related events.

Keyboard and mouse events are pre-defined by Wired for Sound and cannot be edited by the user. Also, note that the media elements you attach to keys will not play while you're typing in a DOS session.

See [Attaching Sounds, Videos, and Pictures to Windows Events](#) for more details.

Startup Events

Wired for Sound allows you to attach media elements to Windows startup and exit. Although Windows 3.1 allows you to specify a startup and exit waveform sound, Wired for Sound allows MIDI sounds, videos, and pictures to be played at startup and exit as well.

Note that the startup event will play when Wired for Sound is loaded, even if this is long after Windows has loaded. It is intended to be used in conjunction with auto-loading Wired for Sound. See Preferences.

Note: In general you should not enable both Wired for Sound's startup and exit events and those in Windows 3.1; otherwise, they may conflict.

See Attaching Sounds, Videos, and Pictures to Windows Events for more details.

Icon Events

Wired for Sound allows you to attach media elements to Program Manager , Norton Desktop for Windows 2.x, and PC Tools Desktop for Windows 1.x icons. The media elements will play when you double-click on the icon to launch an application. First, you must create an Icon event. Then, you attach the desired media elements to the icon event. You can use the Sound... option in the Program Manager's File menu to simplify this task. Note that events for many popular icons have already been created.

If you have enabled Icon events, Wired for Sound will enable Icon Events for a given shell program only if it is loaded before Wired for Sound is loaded. If no compatible shells are loaded and icon events are enabled, Wired for Sound will attempt to load Program Manager.

See Creating Icon Events, Editing Icon Events, and Attaching Sounds, Videos, and Pictures to Windows Events.

Note

Icon events are *not* stored in association files. Thus, when you create a new Icon event, all your association files will be able to have media elements attached to them.

Creating Icon Events

With Wired for Sound you can create Icon events either by using Wired for Sound or from the Program Manager's File menu. Wired for Sound must be loaded in either case.

Using Program Manager

1. Select the icon you'd like to attach media elements to.
2. Choose Sound... from the File menu in Program Manager. Note that this option will be grayed out if you are doing something else in Wired for Sound, such as editing other events.
3. Enter a description for the new event, or accept Wired for Sound's suggestion. Press OK.
4. Choose a sound to attach to the icon from the list of sounds and press OK. You can change other Icon event associations before pressing OK if you wish.

Once you have created an Icon event for a given icon, you can use the Sound... option to change the media elements attached to the icon.

Using Wired for Sound Directly

1. Click on the Icon... event button, or choose Icon Events... from the Edit menu.
2. Click on the Add... button to open the Add Icon Event dialog.
3. Enter the icon's description. This is the text which appears below the icon in the shell (i.e. Program Manager).
4. Enter a description for the event. By default, the description will be the same as the Icon's text, followed by "Icon." Note if you change the description of the event later, media elements attached to it in other association files no longer will be.
5. Click OK to close the Edit Icon Event dialog.
6. You can create or edit more icon events. When done, press OK.

Editing Icon Events

1. Click on the Icon... event button, or choose Icon Events... from the Edit menu.
2. Choose an event from the list and click on the Edit... button to open the Edit Icon Event dialog.

You can change the event and icon description for the event as described in [Creating Icon Events](#).

Note

If you change the description of the event, the media elements which were previously attached to it in other association files will no longer be. You will have to re-attach media elements to the event in all other association files.

Window Events

Wired for Sound allows you to attach media elements to user-specified Windows events. These events can be associated with virtually any application window or dialog box and any of several window actions. Wired for Sound comes with several pre-defined Window events for popular Windows applications and common dialog boxes. To attach media elements to other windows or dialog boxes, you must first create a Window event. Then, you attach one or more media elements to the Windows event.

See Creating Custom Window Events, Editing Window Events, and Attaching Sounds, Videos, and Pictures to Windows Events for more details.

Note

Window events are *not* stored in association files. Thus, when you create a new Window event, all your association files will be able to have media elements attached to them.

Editing Window Events

1. Click on the Window... event button, or choose Window Events... from the Edit menu.
2. Choose an event from the list and click on the Edit... button to open the Edit Window Event dialog.

You can change the Windows Parameters for the event as described in [Creating Window Events](#).

Note

If you change the description of the event, the media elements which were previously attached to it in other sound association files will no longer be. You will have to re-attach media elements to the event in all other association files.

Creating Window Events

Wired for Sound lets you create Window events by clicking on the [speaker button](#) which appears in the title bar of the window or by using the Sound menu item in the system menu of the window. If no speaker button or sound menu item appear, you may still be able to attach a sound to the window by creating it using Wired for Sound directly. You can also create more complicated and powerful custom window events by using Wired for Sound directly.

Creating Window Events Using the Speaker Button

1. Display the window you'd like to attach media elements to.
2. Click on the speaker button. If no button appears, Wired for Sound may still allow you to attach sounds to it by creating an appropriate custom window event directly.
3. Wired for Sound will choose a default description of Open *window name* by default. You can edit the event to change this or the event's action if you desire--see [Editing Window Events](#) for more details.
4. Attach one or more media elements, as described in [Attaching Sounds, Videos, and Pictures to Windows Events](#), and press OK.

Creating Window Events Using the Sound Menu

1. Display the window you'd like to attach media elements to.
2. Open the window's system (or control) menu. This is the small square in the upper-right corner of most windows. Choose Sound from this menu. A sub-menu will list the possible [Window actions](#) your event can use. The sound will play when you perform one of the listed actions to the window. If no Sound menu appears, Wired for Sound may still allow you to attach sounds to it by creating an appropriate custom window event directly.
3. Wired for Sound will choose a default description of *actionname window name* by default. You can edit the event to change this or the event's action if you desire--see [Editing Window Events](#) for more details.
4. Attach one or more media elements, as described in [Attaching Sounds, Videos, and Pictures to Windows Events](#), and press OK.

Creating Window Events Using Wired for Sound Directly

1. Click on the Window... event button, or choose Window Events... from the Edit menu.
2. Click on the Add... button to open the Add Window Event dialog.
3. Make sure that the window is visible. Press the Pick Window... button and point to the title bar or frame of the relevant window. A rectangle should appear around it. If not, then either the mouse isn't pointing to it, the mouse is pointing to a window *within* the desired window, or it's a window Wired for Sound cannot attach media elements to.
4. You may want to further specify one or more [window parameters](#). If you don't know the value of a parameter, accept the value Pick Window put there.

See [Window Parameters](#) for more information on choosing these parameter values.

5. Choose which [Window action](#) you'd like the event to apply to. Each [window event](#) must have a window action, and you must create a separate Custom window event for each window action you want to attach media elements to.

See [Window Actions](#) for more information on selecting an appropriate window action for your event.

6. Click OK to close the Edit Custom Window Event dialog.
7. You can create or edit more custom events. When done, press OK.

See [Window Event Examples](#) for examples of different Window events you can create.

See [Attaching Sounds, Videos, and Bitmaps to Windows Events](#) for details on attaching media elements to your new Window events.

See [Windows and Dialogs](#) for more in-depth information.

Window Parameters

Window parameters let you specify which window you want a Window event to refer to. There are four parameters which Wired for Sound uses:

Parameter	Description
<u>Window Title</u>	The text which appears in the title bar, or caption, of the window.
<u>Message Text</u>	Any text which always appears in a dialog box. This setting is mainly used to distinguish between dialog boxes without captions.
<u>Window Class</u>	An internal value which determines which type a window is. Usually, windows which have similar functionality in the same application (such as the groups in Program Manager) have the same class.
<u>Application Name</u>	The name of the application which is associated with the window.

Window parameters can be specified by typing the desired values in the Add/Edit Custom Window Event dialog or by using Pick Window.

Window Title

This window parameter is the text which appears in the title bar of a window or dialog box.

Notes

Some windows and dialogs have no title bar. This parameter should be blank or set to [Any] for them.

Many applications change the value of their title bar to reflect the currently loaded file. Enter only the text at the beginning of the title which is always the same. For instance, for Windows Write, specify only "Write", not "Write - FILE.WRI". When using Pick Window, the filename portion of the title will be truncated.

Message Text

This window parameter is useful when different dialog boxes have the same window titles. The message text can be any text which always appears when a dialog box is displayed.

Notes

It is best if this text is not user-editable text, such as a file name, since such text may not always be the same when the dialog is displayed. Editable text is usually displayed with a rectangle frame around it.

You need only enter the first few characters of the dialog's message text, as long as this is sufficient to distinguish the dialog box uniquely.

Window Class

This window parameter is an internal value which Windows uses to determine a window's behavior. If the window title and message text are not sufficient to uniquely determine the window, this parameter often will be. It can also be useful when defining Custom window events which refer to many windows.

Notes

Specifying this parameter works best with application windows, as most dialog boxes (but not those in Microsoft Word and Excel!) have the same class (#32770). The class name is chosen by the developer and is not usually visible by the user, so it is best to use the Pick Window feature to determine this value.

Application Name

If two applications have otherwise similar or identical dialog boxes, you can use this window parameter to distinguish between windows. It is also useful for creating Custom window events which refer to all windows an application creates.

Notes

Enter the full name of the application, such as EXCEL.EXE or WRITE.EXE. Some windows are created by dynamic-link libraries, which will have a .DLL extension.

Applications can display windows or dialogs by invoking other applications. Use Pick Window to verify which application has created the dialog or window.

Pick Windows

Often it is easiest to let Wired for Sound determine the window parameters.

To do this, the window or dialog box should be displayed before creating or editing the Window event. When the Add or Edit Window Event dialog box appears, press the Pick Window... button. As you move the mouse pointer over various windows, the window parameters will change to reflect the window you're pointing to. The window frame will also be inverted. When you've selected the appropriate window, click on it. Since there are a lot of nasty things on the screen that Windows also knows as windows, it's best to click on the title bar of the window you want to specify.

See also: Windows and Dialogs

Window Actions

Window actions define what must happen to a Window for the media elements attached to the Window event to play.

Wired for Sound lets you specify the following basic window actions:

Action	Description
Open	This action happens when a window or dialog box is initially displayed, or whenever a minimized icon is restored.
Close	This action happens when a window is closed.
Minimize	This action happens when a window is minimized.
Maximize	This action happens when a window is maximized.
Restore	This action happens when a maximized window is returned to its normal size. If you want media elements to play when you restore a window that is an icon, specify "Open" as the action.
Move	This action happens when you click on the title bar of a window to begin moving it.
Moving	This action happens as you move a window around the screen with the mouse. The sound will be repeated as the mouse moves.
Moved	This action happens when you release the mouse after moving a window.
Resize	This action happens when you click on the border of a window to begin resizing it. The window must have a resizable ("thick") border.
Resizing	This action happens as you resize the window with the mouse. The event repeats as the mouse moves.
Resized	This action happens as the mouse is released after resizing a window.

Other, more advanced actions, may also appear and are discussed in Advanced Actions.

Advanced Actions

The following are advanced actions which are included mainly for users who really want control of their Windows sound environment. They are also used in several of the pre-defined Window events. They can interact in strange ways with other actions, as indicated. Please refer to [Windows and Dialogs](#) for more information on the terminology used here.

Action	Description
Click-Activate	This action occurs whenever a window is made the active window by clicking on it with the mouse.
MDI Activate	This action occurs when a multiple-document interface child window, such as a Program Manager group or File Manager directory window, is activated. The actual child window (i.e. Program Manager group window) should be specified in the Window event. This action could interfere with a Click-Activate action for the window which contains the MDI child window if you click on the child to activate the application.
Vertical Scroll and Horizontal Scroll	These actions occur when a vertical (or horizontal) scrollbar within a window is scrolled. Note that for some applications, you cannot attach Window events to the window which contains the scroll bars.
MDI Tile, MDI Cascade, and MDI Arrange Icons	This action occurs when an application tiles, cascades, or arranges its MDI children. You should specify the MDI Client window (class MDIClient) when using this action.
Open Popup Menu	This action occurs when a pop-up menu is displayed in the given window. Be sure to specify the top-level window with the menu bar in it.
Select Menu Item	This action occurs when a menu item is selected. Be sure to specify the top-level window with the menu bar in it. This action interferes with the Open Popup Menu action for the same window since a menu item is selected when you first open a popup window.

Windows and Dialogs

This topic is intended for advanced users who want to have more control over Window events.

Although windows and dialog boxes often appear different and are used for different purposes, Windows considers them and many other things as simply "windows". Wired for Sound's Window events allow you to attach sounds to any window which has a border, old-style modal dialog frame (for captionless dialogs), caption, modal dialog frame, or is a special system class. The [Pick Window](#) feature will highlight only windows which meet these criteria.

Multiple-Document Interface (MDI) Applications

Programs like the Windows Program and File Managers display small windows within a larger window. These small windows (i.e. the Program Manager groups and group icons) are called MDI child windows. The application's main window (the one with the menu) is called the frame window. A third window, which forms the background for the MDI child windows, is the MDI Client window. Window events can be created for any of these windows, depending on the action. Minimize, maximize, show, destroy, and move can be used with both frame and MDI child windows. Click-activate works only for top-level frame windows, while MDI Activate works only with the MDI child windows.

Special Window Classes

Many system windows have special class names. Many of the pre-defined events use these special classes.

Class	Windows that use it
#32768	Pop-down menus. This class can be used with the Show action to produce a sound when a menu is accessed. Alternatively, you could use the Open Popup Menu action with [Any] window and class.
#32769	The desktop (background) window. We know of no useful event which uses this class.
#32770	Standard dialog boxes and message boxes. This class is used for many Open and Close events
#32772	The caption of a minimized window. We know of no useful events which use this class.
ListBox	Standard list boxes. This class is used with the Vertical Scroll action for the Vertical Scroll List Box event.
ComboLBox	The drop-down list of a combo control. This typically appears when you push the down arrow to the right of the combo control. It is used with Vertical Scroll like the ListBox class.
MDIClient	The background window for MDI applications. Specify this window for events which use the MDI Tile, MDI Cascade, and MDI Icon Arrange events.

Microsoft Applications

You will probably have to create Window events to attach sounds to dialog boxes in Microsoft applications since they don't use standard Windows dialog boxes. Also, Word and Excel are not standard Multiple-Document Interface applications.

Also note that the buttons in some Microsoft applications are not true Windows buttons, so Button events will not work for them.

Window Event Examples

To help you get started with creating Window events, the following examples are provided. You should already be familiar with the basic steps used to create Window events. You may also want to examine the pre-defined Window events--simply select them and choose Edit (but be sure not to make any changes unless you know what you're doing!)

Creating an Event for Any Window Being Minimized

1. Enter the Add Window Event dialog box.
2. Enter "[Any] Minimized" as the description.
3. Enter "[Any]" for the other four window parameters.
4. Choose Minimize as the Window Action.

Note

Wired for Sound searches custom events in reverse alphabetical order according to their description name. To prevent a custom event with many [Any] parameters from preventing more specific Window events from working, the description begins with "[Any]", which will cause it to be found after other events. In general all descriptions beginning with punctuation will be searched after those beginning with alphabet characters.

Creating an Event for all Windows Write Dialogs

1. Enter the Add Window Event dialog box.
2. Enter "Any Write window opened" as the description.
3. Enter "[Any]" for all but the Window Class and Module Name window parameters. Enter "WRITE.EXE" for the Application Name and "#32770" for the Class Name.
4. Choose Open as the Window Action.

Note

Most dialog boxes have a class name of "#32770".

Creating an Event for File Manager Being Opened

1. Enter the Add Window Event dialog box.
2. Enter "File Manager Opened" as the description.
3. Enter "File Manager" for the Window Text and "WINFILE.EXE" as the Application Name. The other two parameters should be "[Any]".
4. Choose Open as the Window Action.

Notes

This event could interfere with an Icon event attached to File Manager's icon. This type of event is useful when you want a sound to be played when an application is launched even if you don't use Program Manager or one of the other supported shells.

You could enter the parameters by running File Manager, clicking on Pick Window..., and clicking on the File Manager's title bar. This will also enter the class parameter, which isn't necessary since the title and application name uniquely identify the File Manager's main window. See [Windows and Dialogs](#) for more about File Manager and similar applications.

Creating an Event for Closing the Windows Notepad

1. Enter the Add Window Event dialog box.

2. Enter "Notepad Closed" as the description.
3. Enter "Notepad" for the Window Text and "NOTEPAD.EXE" as the Application Name. The other two parameters should be "[Any]".
4. Choose Close as the Window Action.

Notes

Notepad, like many Windows applications, sometimes puts the name of the file being edited in its title bar. You should only specify the part of the title which doesn't change for the Window Title.

If you use the Pick Window... feature, make sure you don't choose the text area of the Notepad window. It is a separate window which has "Edit" as its Class Name and "USER.EXE" as its Application Name.

Windows Event

A Windows event can be caused by either user or system action. Examples include displaying windows or dialog boxes, clicking on a button or icon, and system errors.

Window Action

A window action is caused whenever a window is moved, minimized, maximized, hidden, opened, or destroyed.

Window Event



A Window event can be created to attach a sound to Windows events which aren't already defined in Wired for Sound. These can include events generated by specific Windows applications.

System Error Event



A System Error event is a Windows event which happens when an application does something Windows doesn't like. An example is the UAE or Application Error.

Button Event



A Button event is a Windows event which corresponds to pressing a button in a window or dialog box.

Icon Event



An Icon event can be used to attach media elements to Program Manager icons, or the icons of a compatible, supported Windows shell, such as Norton Desktop for Windows. The media elements for an icon event will play when you double-click on or open the icon.

Startup Event



A startup event can be used to attach media elements to Windows startup and exit.

Keyboard Event



A keyboard event can be used to attach media elements to various keys on your keyboard.

Window Parameter







A window parameter is a value which Wired for Sound uses to determine which window or dialog box a custom window event refers to.

Association File

An association file determines which media elements, if any, are played when a Windows event happens.

Event Button

The six buttons at the top of Wired for Sound's window are the event buttons. These can be used to attach media elements to Windows events.

Button	Description
	This button accesses the <u>Button events</u> .
	This button accesses the <u>System Error events</u> .
	This button accesses the <u>Icon events</u> .
	This button accesses the <u>Window events</u> .
	This button accesses the <u>Keyboard events</u> .
	This button accesses the <u>Startup events</u> .

Speaker Buttons



Speaker buttons are small buttons which appear in the title, or caption, bars of many windows when Wired for Sound is loaded. Pressing one allows media elements to be attached easily to most windows.

Media Element

A media element is any multimedia file, such as a sound file (waveform or MIDI), a video clip (such as AVI), or a Windows bitmap. If you have the speech synthesis module, text files are also treated as media elements--they will be read by the synthesis engine when played. Talking Clock in general allows any type of media element to be specified when you can choose a media element for a chime or appointment.

