Introduction

This article explains how to use the mouse and keyboard library that I have created. This library consists of two things: global mouse and keyboard hooks, and global mouse and keyboard simulators.

The global hooks contain a set of events that follow the .NET event model, so they should be very simple to use if you've done anything with events before.

The simulators will actually simulate mouse movements, clicks, keyboard presses, etc. This can be useful for macro recording (which is one of the sample projects), and of course, messing with your friends. :)

Background

I've found a lot of hook and simulator code out there, but a lot of it was not very organized, and a bit hard to use. The goal here is to make a very simple and easy-to-use library for mouse and keyboard operations not natively supported by .NET.

Global Hooks

This section will explain how to use the global hooks which can capture mouse and keyboard events from any application. These events are very similar to the ones that appear on Windows controls, so they should look familiar.

Using the mouse hook:

Hide Copy Code

```
// Create the mouse hook
MouseHook mouseHook = new MouseHook();

// Capture the events
mouseHook.MouseMove += new MouseEventHandler(mouseHook_MouseMove);
mouseHook.MouseDown += new MouseEventHandler(mouseHook_MouseDown);
mouseHook.MouseUp += new MouseEventHandler(mouseHook_MouseUp);
mouseHook.MouseWheel += new MouseEventHandler(mouseHook_MouseWheel);

// Start watching for mouse events
mouseHook.Start();

...

// Stop watching (don't forget to do this before closing application!)
```

```
// Stop watching (don't forget to do this before closing application!)
mouseHook.Stop();
```

Using the keyboard hook:

Hide Copy Code

```
// Create the keyboard hook
KeyboardHook keyboardHook = new KeyboardHook();

// Capture the events
keyboardHook.KeyDown += new KeyEventHandler(keyboardHook_KeyDown);
keyboardHook.KeyUp += new KeyEventHandler(keyboardHook_KeyUp);
keyboardHook.KeyPress += new KeyPressEventHandler(keyboardHook_KeyPress);

// Start watching for keyboard events
keyboardHook.Start();

...

// Stop watching (don't forget to do this before closing application!)
keyboardHook.Stop();
```

Note: When you are setting the events, Visual Studio will name and create a blank method for you. You only need to type this much on the event ...

Hide Copy Code

```
keyboardHook.KeyDown +=
```

...and then hit **TAB** two times. Visual Studio will finish the rest of the line, **and** will go out and create the blank method for you. This is a nice feature, and saves a lot of time; use it!

Simulators

This section will explain how to use the the mouse and keyboard simulators to simulate mouse clicks and keyboard key presses. Both the `KeyboardSimulator` and `MouseSimulator` classes are `static`, so they are pretty simple to use.

Simulating mouse events:

Hide Copy Code

```
// Press Left Mouse Button Down
MouseSimulator.MouseDown(MouseButton.Left);

// Let Left Mouse Button Up
MouseSimulator.MouseUp(MouseButton.Left);
```

```
// Press down and Let up Left Mouse Button
// (equivalent to two lines above)
MouseSimulator.Click(MouseButton.Left);

// Double click Left Mouse button

// (equivalent to two Click()s above)
MouseSimulator.DoubleClick(MouseButton.Left);
```

The code above is used to simulate pressing and letting up a certain mouse button. The one parameter is just an enumeration of the three possible mouse buttons: `Left`, `Right`, and `Middle`.

Moving the mouse position:

Hide Copy Code

```
// Move mouse cursor to Top Left of screen
MouseSimulator.X = 0;
MouseSimulator.Y = 0;

// Move the mouse cursor to the right by 20 pixels
MouseSimulator.X += 20;
```

The `X` and `Y` above are properties. You can use them to get the current position of the mouse cursor, or you can set them to move the mouse cursor to a new location.

Keyboard simulators:

Hide Copy Code

```
// Press the A Key Down
KeyboardSimulator.KeyDown(Keys.A);

// Let the A Key back up
KeyboardSimulator.KeyUp(Keys.A);

// Press A down, and let up (same as two above)
KeyboardSimulator.KeyPress(Keys.A);
```

The code above will simulate keyboard key presses. You can press a key down (first line), which doesn't let it up yet. The second line, `KeyUp`, will release a key that has been pressed down, the third line with do both steps in one shot.

I also included some standard keyboard shortcuts. These can all be done with the code above, but it simplifies it a bit, and makes the code a bit more readable and obvious.

Hide Copy Code

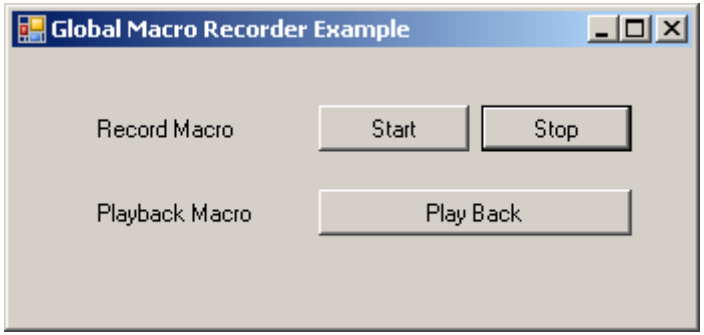
```
// Simulate (Ctrl + C) shortcut, which is copy for most applications
```

```
KeyboardSimulator.SimulateStandardShortcut(StandardShortcut.Copy);

// This does the same as above
KeyboardSimulator.KeyDown(Keys.Control);
KeyboardSimulator.KeyPress(Keys.C);
KeyboardSimulator.KeyUp(Keys.Control);
```

The code above does the exact same thing twice, except the first is a bit shorter and more obvious.

A Sample Application: Global Macro Recorder



A macro recorder is a great example for this library, since we can use the hooks to record the macro, and the simulators to play back.

Check out the Macro project in the downloadable source code.

Future Additions / Revisions

I'm going to edit and add to this library over time as I get feedback. If you think something should be added, changed, or find a problem, please post in the comments section, and I'll do what I can. Thanks. :)