# Script the Clicking – Automating Windows with MKS Toolkit

Odds are good that if you use MKS Toolkit, you are familiar with using shell scripts to automate various command line processes. But did you know you can also automate Microsoft Windows applications? Well, you can, thanks to the MKS Toolkit **`winctrl`** and **`sendevent`** utilities. The **`winctrl`** utility, as its name implies, lets you control open windows. With it, you can close, move, or give focus to a window, or simply retrieve or set the window's attributes. The **`sendevent`** utility, not surprisingly, sends events to the window which currently has focus. These events can be keystrokes, mouse movements, or mouse button pushes. You can combine these two utilities to automate many Windows-based tasks. Two possible uses are the automated testing of Windows applications and the inclusion of Windows tasks as part of a shell script.

To demonstrate these utilities, we are going to take a simple task (sending an e-mail message with Microsoft Outlook) and show how **`winctrl`** and **`sendevent`** can be used to automate that task.

## THE BASIC TASK

As mentioned above, the task we will be automating is the sending of an e-mail message with Microsoft Outlook. In particular, we will be sending the message "This is a test" to a user named `nobodyatall@mks.com`.

The first thing we do is break this task down into a number of distinct steps. This makes it easier to see just what **`winctl`** and **`sendevent`** need to do. Here are the basic steps for sending our e-mail message:

1. Launch Microsoft Outlook.
2. Give focus to the Microsoft Outlook window.
3. Launch the dialog for sending a new mail message. This dialog now has the focus.
4. Fill in the fields of the message.
5. Move the mouse to the **Send** button and click the left mouse button to send the e-mail message.
6. Close the Microsoft Outlook window.

Before proceeding with the discussion of how to implement these steps using **`winctrl`** and **`sendevent`**, it should be noted that this example uses Microsoft Outlook 2002 (from Office XP). However, the techniques are similar for any version of Outlook, although the specifics may change in a few places.

## AUTOMATING THE TASK

Now that we've broken out task down into the seven steps described earlier, it is time to actually start looking at how we can implement these steps using **`winctrl`** and **`sendevent`**. Note that while we'll be looking at each step individually, each of these steps should be considered to be a part of a MKS KornShell script. Why? Because if commands were issued directly from the command line instead of a script, every command would need to first transfer focus back to the window we wanted to work in.

### Step 1: Launch Microsoft Outlook

Let's look at two methods of launching Microsoft Outlook. The first is to simply launch the application by giving its full path:

```
"c:/program files/microsoft office/office10/outlook" &
```

The precise path given here is the path on my machine to Microsoft Outlook. The path on your system may vary. The & following the path name instructs the MKS KornShell to simply launch the program and then return for more commands. Without the &, the shell would wait for the Outlook window to be closed before looking for more commands.

The other method for launching Microsoft Outlook is to simply send the keystroke that launches the default mail program. Of course, this only works if you have Outlook set as your default, but for the purposes of this discussion, we'll assume you do. So, just what keystroke does launch the default mail program? The answer to that is the keyboard event known internally to Windows as `VK_LAUNCH_MAIL`. If you have ever seen one of those Internet keyboards which have a special key for starting your mail program, this is the keyboard event sent when you press this key. So, to launch Microsoft Outlook, we simply need **`sendevent`** to "press" the `VK_LAUNCH_MAIL` key (even if there is no actual key on your keyboard to launch mail). We can do this with the following command:

```
sendevent "<launch_mail><LAUNCH_MAIL>"
```

What happened to the `VK_` prefix, you ask? Well, because all keyboard events begin with `VK_` and **`sendevent`** uses `< >` only for keyboard events, the `VK_` is redundant and sendevent does not require you to specify it. And why is the keyboard event specified twice, once in lowercase and once in upper? In truth, hitting a key is two events: pressing the key down and releasing it. When keyboard event is specified in lowercase, it is the equivalent of pressing the key down. Specifying the event in uppercase is the equivalent of releasing the key. So, in this example, we are pressing then releasing the virtual key.

## Step 2: Give Focus to Outlook Window

Okay, now that we have launched Microsoft Outlook, we're going to want to give the focus to that application window. When a window has focus, all **`sendevent`** commands affect that window. To give focus to a window, you use the **`setfocus`** sub-command of **`winctrl`**, as in:

```
winctrl "Inbox – Microsoft Outlook" setfocus
```

The window title that you use — in this case, **Inbox – Microsoft Outlook** — depends upon which Outlook window you have set as your default. For example, if you had it set to show your calendar window, you would use **Calendar- Microsoft Outlook** instead.

## Step 3: Launch Mail Message Dialog

The next step in our process is to launch the dialog for creating a new mail message. There are several ways to do this from within Outlook. The easiest for automation purposes is to type `Ctrl-Shift-M`. This key combination works from any Outlook window. To send this combination, you can use:

```
sendevent "<ctrl><shift><m><M><SHIFT><CTRL>"
```

which means "press the Ctrl key, press the Shift key, press the M key, release M, release Shift, release Ctrl". This may seem a bit verbose for sending one key combination. Fortunately, **`sendevent`** allows a couple of shortcuts to make sending strings of characters easier. For example, for most character keys, just entering the character is the equivalent of pressing and releasing the key. For example, `m` is the equivalent of `<m><M>`, so we can simplify our command to:

```
sendevent "<ctrl><shift>m<SHIFT><CTRL>"
```

Of course, in a similar manner, `M` is equivalent to `<shift>m<SHIFT>`, meaning we can simplify the command even more to:

```
sendevent "<ctrl>M<CTRL>"
```

When executed, this command launches the dialog for creating a new mail message. As is usual when a new dialog is launched, that dialog has the focus.

## Step 4: Fill in Message Fields

*Figure 1: Mail Message Dialog* shows the new mail message dialog as it appears on my system when I type `Ctrl-Shift-M`. Of course, it may look slightly different on your system if you have different toolbars displayed or have the toolbars arranged differently, but for our purposes, we're going to be assuming the version shown here. When this dialog appears, the cursor is automatically placed in the **To...** field. So, to enter the address where we want to send our message (`nobodyatall@mks.com`), we simply need to use:

```
    sendevent "nobodyatall@mks.com"
```

Once we have entered that field, we need to move the cursor to the next field (in this case, the **Cc...** field). To do so, we would normally hit the Tab key. Using **`sendevent`**, this would be:

```
    sendevent "<tab><TAB>"
```

which, of course, is equivalent to pressing the Tab key and then releasing it.  In fact, we can combine this with the previous sendevent command, giving us:

```
    sendevent "nobodyatall@mks.com<tab><TAB>"
```

When this command is executed, the text is entered in the **To...** field and the cursor is moved to the **Cc...** field. However, for this message, we are not copying anyone so we want to leave this field blank and move on to the **Subject:** field. We can do this with:

```
    sendevent "<tab><TAB>"
```

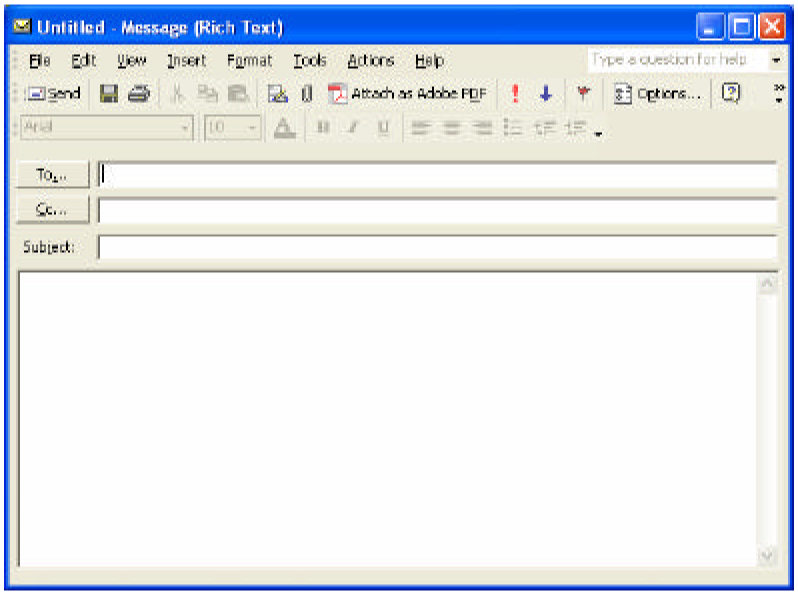

***Figure 1: Mail Message Dialog***

With the cursor in the **Subject:** field, it is time to enter the subject of our e-mail message. For reasons you'll see in the next section, let's go with something simple like "Test". We can enter this subject line with:

```
    sendevent "Test<tab><TAB>"
```

This also moves the cursor into the message area of the dialog. We can now enter our message of "This is a test" with the command:

```
    sendevent "This is a test"
```

Our mail message has been entered and is ready to be sent.

## Step 5: Click the Send Button

There are actually (at least) two ways to send our e-mail message. The first is simply to type `Alt-S` and this could be done with the command:

```
    sendevent "<alt>s<ALT>"
```

However, we've already seen how to send a key combination like that, so instead, let's look at the second method: clicking the **Send** button. The first thing we need to do is move the mouse to the button. But where is the button? Well, we can roughly tell where it is on the mail message dialog. A quick eyeballing of the situation shows that if we were to place the mouse about 20 pixels from the left edge of the dialog and about 75 pixels down from the top edge, it would be over the **Send** button. But before we can place the mouse relative to the dialog, we need to determine where the dialog appears on the screen. Fortunately, **`winctrl`** helps us out here. The commands:

```
    winctrl "Test" getleft
    winctrl "Test" gettop
```

will return the horizontal and vertical co-ordinates, respectively, of the upper left corner of the window named **Test**, which is the name of the mail message dialog. Where did we get **Test** from? Well, as soon as we entered "Test" as the subject line, that became the name of the window (technically, the first part of the name, but that's good enough for **`winctrl`**).

Because we are putting these commands in an MKS KornShell script, we can use the shell's `$(...)` construct to include these values directly in a **`sendevent`** command to move the mouse to the upper right corner of the mail message dialog, as follows:

```
sendevent "{a$(winctrl "Test" getleft),$(winctrl "Test" gettop)}"
```

The a that precedes the co-ordinates within the `{...}` construct tells **`sendevent`** that we are giving absolute co-ordinates (that is, the position of the mouse relative to the entire desktop). You'll notice that the left and right curly brace, { and }, are used to represent mouse commands similar to the way that < and > are used for keyboard commands.

Now that we have placed the mouse at the upper left, we can use the command:

```
sendevent "{20,75}"
```

to move the mouse 20 pixels to the right and 75 pixels down (that is, over the **Send** button). All we have to do now is click on the button. We can do this with:

```
sendevent "{l}{L}"
```

This is similar to pressing and releasing a key. `{l}` is equivalent to pressing the left mouse button while `{L}` is equivalent to releasing it. Once the button has been pressed, our message is sent and we have almost completed our task.

## Step 6: Close Outlook Window

So far, we have launched Microsoft Outlook, created and entered a new mail message, and sent that message to `nobodyatall@mks.com`. We have only one thing left to do: shut down Outlook. To do this, we simply need to close the **Inbox - Microsoft Outlook** window with the following **`winctrl`** command:

```
winctrl "Inbox - Microsoft Outlook" close
```

Again, the name of this Window may be different on your system if Microsoft Outlook launches with a different default window.

## The MKS KornShell Script

Now that we have looked at all the individual steps that we are going to use in our script, let's take at a look at the final version. *Figure 2: The MKS KornShell Script* shows the completed product. In addition, to the commands discussed earlier, you will notice a few additional commands of the form:

```
sendevent "[2000]"
```

This command tells the script to pause for the specified number of milliseconds before continuing. The main purpose for such commands in our script is to allow time for a window or dialog to launch. Again, these are the pauses that work on my system; your system may differ.

```
sendevent "<launch_mail><LAUNCH_MAIL>"
sendevent "[5000]"
winctrl 0x0027014c setfocus
winctrl "Inbox - Microsoft Outlook" setfocus
sendevent "<ctrl>M<CTRL>"
sendevent "[2000]"
sendevent "nobodyatall@mks.com<tab><TAB>"
sendevent "<tab><TAB>"
sendevent "Test<tab><TAB>"
sendevent "This is a test"
sendevent "{a$(winctrl "Test" getleft),$(winctrl "Test" gettop)}"
sendevent "{20,75}"
sendevent "{l}{L}"
winctrl "Inbox - Microsoft Outlook" close
```

***Figure 2: The MKS KornShell Script***

## SUMMARY

Well, that's it. Through the simple example of sending an e-mail with Microsoft Outlook, you have been introduced to many of the features of both **`winctrl`** and **`sendevent`**. While you may have no reason to automate the sending of e-mail in this way, it should be easy for you to apply the techniques presented here to automate almost any Microsoft Windows task.

For more information about the MKS Toolkit products please visit http://www.mkssoftware.com and to view the full reference pages for the commands mentioned in this document visit http://www.mkssoftware.com/docs/cmd_index.asp.