

# **SST Programming Software Contents**

[Getting Started](#)

[Simple Commands or Keystrokes](#)

[Multiple Keystrokes/Combined Keystrokes](#)

[Macros](#)

[Advanced Commands](#)

[Shift Modes](#)

[Saving the Profile](#)

[Activating the Profile](#)

[Testing the Profile](#)

**Introduction**

Welcome to the manual for the latest version of our programming software, Saitek Smart Technology (SST). Following comments and feedback from our customers we have made many changes to the software from previous versions and hope that you now find it much easier to use; we also hope that as newer versions are released you will find it becomes more powerful too.

As with previous versions of the programming software it must be pointed out that this software is not needed to make your controller work in the game – that is usually performed by enabling joystick control from within the game itself. The SST software is used to configure the buttons etc. on your controller to perform different actions within your games and add extra functionality.

Most modern games do have their own control configuration screens that perform the same task but some do not and this was one of the original reasons for our creating the programming software. The other reason for using the SST software is that you can increase the number of commands that can be assigned to your joystick due to shift states (Cyborg USB and X36/X45) and multiple modes (X36/X45); these will be explained later in the manual. Before we cover those more advanced features however, we have to show you how to start programming your Saitek controller.

## Getting Started

Profiles are sets of custom instructions that tell an application such as a game how to respond to the operation of the buttons and other controls. Here we will explain how to create your first profiles.

Having installed the SST software and plugged in your controller you should have a small black joystick icon in the taskbar next to your clock. It should look something like this.

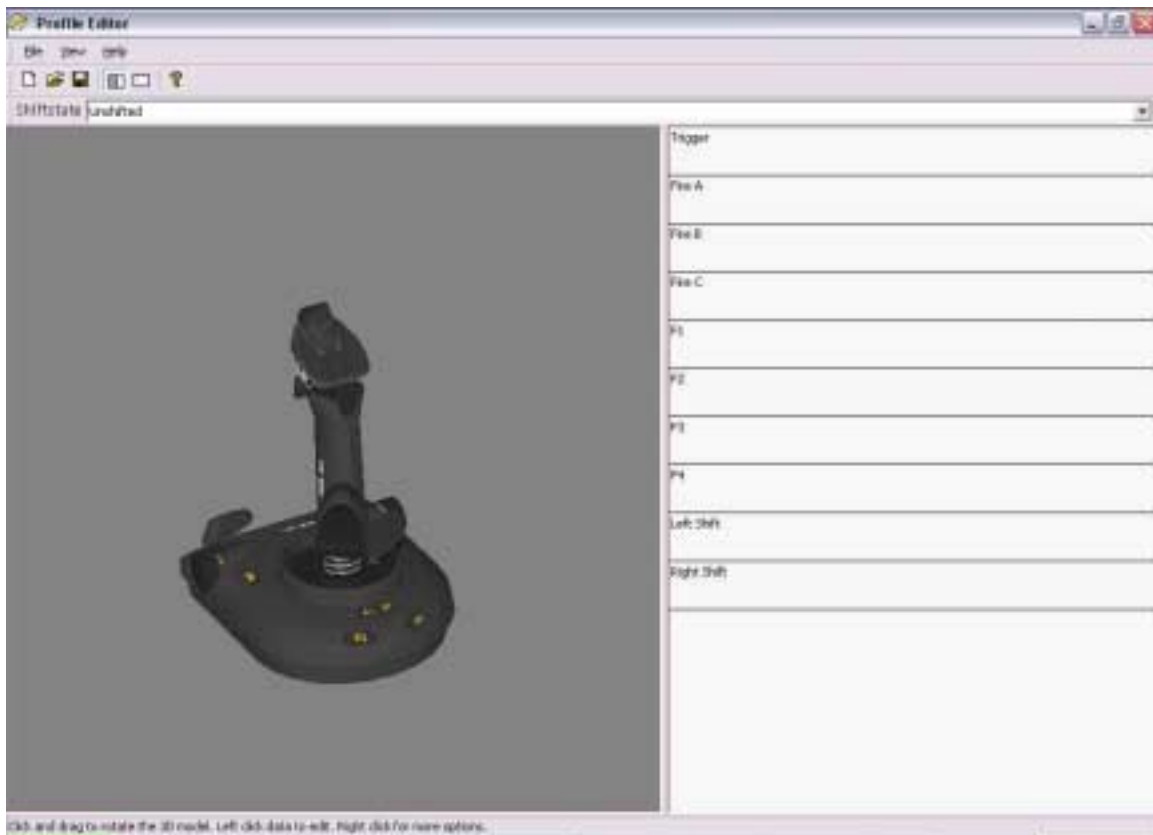


Just a quick note here for any customers who have more than one controller (a Cyborg and an X36, for example) and have them both plugged in, you will have a separate icon for each controller which looks like the controller it refers to – if you hold your mouse pointer over the small joystick icon you will get a small pop-up notice which tells you which controller the icon refers to.

Right-click on the joystick icon and you should get a pop-up menu like this.



Clear Profile is used whenever you want to completely clear a profile from your controller. Clicking **Control Panel** will bring up the test and calibration screens for your controller but for now we want to left click **Profile Editor**. This will present you with the following screen (note that we are using the Cyborg stick in this example but the procedure for programming the Cyborg Force, X36/ X45 and R440 is exactly the same. The only difference is the number of buttons and, in the X36 and X45's cases, the number of modes). This can also be run by double-clicking the Saitek Smart Technology Programming Software link that is on your desktop or by clicking **Start>Programs>Saitek>Profile Editor**.



Those of you familiar with the original SGE programming software will recognise the 3D model of your controller. The major change to the programming software though is the list of buttons down the right hand side of the screen. If you press

one of the buttons on your controller now you will see that it lights up on the 3D model and that the appropriate line in the list of buttons is “highlighted” by a slightly darker background – this makes it easy to see exactly which button you are programming. This is called the 3D View – there is another view you can use and this is outlined further on in the manual.

Every game you have already has most of the in-game commands assigned to various buttons or combination of buttons on the keyboard. For those of you unfamiliar with how this process works, what we are going to do is make the buttons on your controller “pretend” to be buttons on the keyboard.

## Simple Commands or Keystrokes

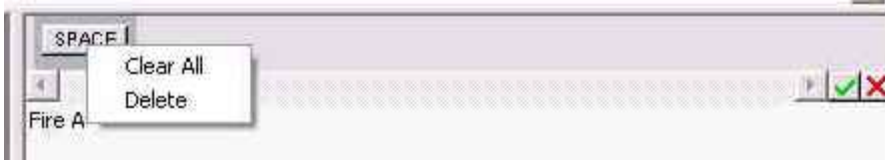
To program a keyboard command to a button is very easy. On the right side of the window in the list of buttons, simply click on the space just beneath the name of the button you wish to program (the mouse cursor will change shape to indicate that you can click it) – in the example below we have left clicked on the space just beneath the trigger.



This gives us a flashing cursor indicating that the software is now waiting for us to input a keystroke. Which keystroke do we input here? As an example, imagine we are programming the controller for a game and we want to use the trigger for firing a weapon. In the game, the key on the keyboard that does this is the Space key. So all we do now is press that key on the keyboard and it will appear in the line onscreen; like so.



If this is okay, then simply click the green tick at the bottom right of the line. If you have made a mistake or wish to clear a keystroke you have put in, then simply right-click on the keystroke you wish to remove and click **Delete** from the drop down list of options. Note that you can also simply left click on the next button you wish to program and this will also save the keystrokes that you just inputted.

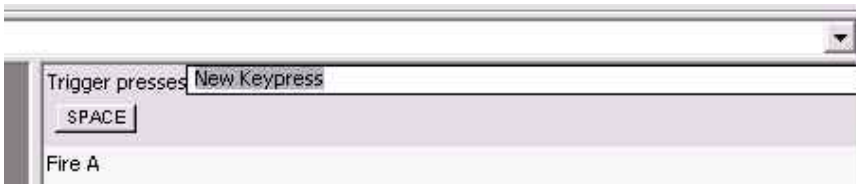


If you wish to clear all the keystrokes from the line then click **Clear All**.

Having decided on your keystroke and left-clicked the green tick, you should have a screen that looks something like this.



As you can see the software is now telling us that the trigger, when pressed, will initiate the Space bar command on the keyboard. This command has been named New Keypress as it was the first new keystroke that we have programmed. If you wish to rename the keystroke to something more recognisable (like Fire Weapon, for example) then this is easily done. Simply point the mouse cursor at the words New Keypress and click the left mouse button once. You will see that the words are now highlighted, as in this picture,



and you can now type the name that you wish to call this command. In this case we're calling it Fire Weapon but of course you can call it whatever is appropriate to the command you are creating. Simply press the **Enter/Return** key after typing your chosen command name in and it will update as seen below.



Congratulations, you have just created the first command for

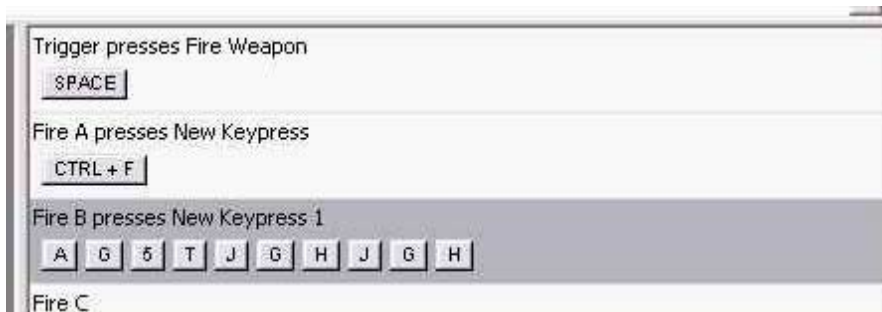
your controller.

## Multiple keystrokes/Combined keystrokes

You are not limited to single keystrokes when programming commands. You can input as many keys into one command as you like or combined keys, like Ctrl+F, for example. The process is the same as inputting single keystrokes – just press the keys you wish to program into the command. In the example below we have inputted a Ctrl+F command.



You can see that it is a combined keystroke because both letters are on the same "key" and there is a plus sign between them. Multiple, single keystrokes would appear like the line for Fire B in this picture.



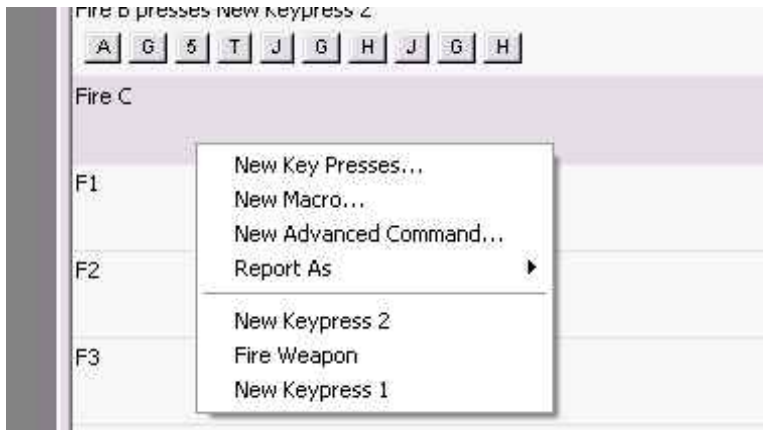
## Macros

Having covered the basic types of keystroke we need to introduce another type of keystroke command – a macro. A macro is a sequence of keystrokes that can be executed with the single press of a button. At first glance you might think that this is exactly the same as the multiple keystrokes, as assigned to the Fire B button in the above example, however this is not the case.

In order for the multiple keystrokes in the above example to actually happen you have to keep the assigned button held down until all of the commands have happened. If the above sequence had been assigned as a macro then you would just have to press the assigned button once and the keystrokes would then input themselves automatically.

You might ask why you need both types of command and the answer is that a macro can't be interrupted. A simple sequence of keystrokes, like the one in the above example, can be interrupted by letting go of the button. Depending on the in-game situation, this could be important.

Creating a macro is similar to creating a keystroke sequence but to access it we have to use the right mouse button. In the below example we have pointed to the Fire C line with the mouse and clicked the right mouse button (right-click)



Before we carry on, let's just take a minute to examine this drop-down menu. As you can see, we have a number of possible choices. The first is **New Keypresses** – left clicking this would enable you to input keystrokes for a new command, just like we have been doing in the above examples.

**New Macro** is the option we will be clicking in a minute...

**New Advanced Command** allows us to do some interesting things involving repeat functions (amongst other things) and we will be covering it later.

**Report As**, when you hold the mouse over it, brings up a list of the different button commands allowing you to make the button pretend to be a different button on the controller, should you so wish.

Underneath these four options you can see listed the commands we have created previously. The New Keypresses are still called that because we didn't rename them but, as you can see, the Fire Weapon command is there. If we wanted the Fire C button (as it is in this case) to also fire the weapons in our hypothetical game, all we would do is left click Fire Weapon and that would assign the same command to Fire C.

However, for now, left click **New Macro** and you will get a small window like this.



All you need to do now is input your keystroke sequence, exactly as you would input it in the game if doing it normally on the keyboard. As you press the keys you want you will notice a number appearing under each key in the Macro Recorder window. This represents the time, in seconds, since you pressed the first key in the macro sequence – obviously therefore the first key in the sequence has 0.00 underneath it. In the example below you can see that there was a 4.75 second wait between pressing the P and S keys.



Left click **OK** when you have finished inputting the macro sequence that you want. Just as with keystrokes, you will notice that the entire sequence of keys is shown in the bar for the button to which you have just assigned this macro. You can also rename this new macro, just as we did with the Fire Weapon command earlier. Just in case you have trouble remembering which commands you have inputted are keystrokes and which are macros, we have slightly changed the way the programming software refers to these commands. As you have seen, the keystrokes commands have the line "Trigger **presses** Fire Weapon"; if it was a macro then it would read "Trigger **plays** Fire Weapon". It's good to remember this simply because of the differences in the way keystroke and macro sequences differ in their implementation as described earlier.

## Advanced Commands

Going back to the drop-down menu that appears when you right-click on a button line, click the **New Advanced Command** option. You will see the following window.



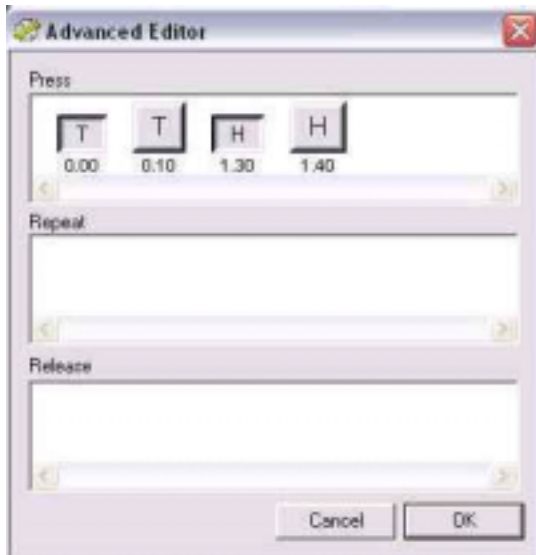


As you can see, this looks similar to the Macro Editor. Each window represents a different state of the button to which you are assigning the Advanced Command. Any keystrokes in the Press column will happen when you press the button the command is assigned to. If you have multiple keystrokes you will need to keep the button held down until the commands have happened – this is exactly the way that normal multiple keystrokes work (i.e. this doesn't function like a macro).

Any keystrokes that are in the repeat column will happen as long as you keep the button they are assigned to held down – again, these will only operate as normal multiple keystrokes.

Any commands in the release column will happen as soon as you release the button to which the Advanced Command is assigned. The difference with the repeat column, though, is that any multiple keystrokes will act like a macro.

You will notice that unlike the other keystroke input windows each key, when pressed, places two instances of the key in the command input window – like below.



This is because when you press a key on a keyboard it actually produced two signals – one when you press it and another when you let go. In the above example you can see that the T and H keys were held down for 0.10 of a second. Just like in the Macro Recorder, the numbers underneath the keystrokes represent the time in seconds from when you pressed the first key in the sequence; in the Advanced Editor though you can adjust these timings. Simply point the mouse cursor at the time you wish to adjust and you will notice that the cursor changes to a clock with two arrows either side of it. Now click and hold down the left mouse button and drag the mouse right to increase the time or left to decrease. Once you are happy with the timing just let go of the mouse button and the new timing will be saved.

Therefore with the Advanced Editor you could set, for example, a command that launches a missile when you press the button. It might then switch to a camera view of the missile and then 4.5 seconds later switch to a camera view of the enemy. When you let go of the button it could switch back to your cockpit view. Obviously this is just a hypothetical example but it gives you an idea of the possibilities open to you with the Advanced Editor.

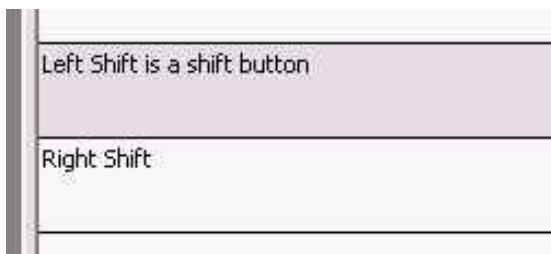
## Shift Modes

With the exception of the Cyborg 3D Force Stick and the R440 Wheel, the controllers supported in this software release all have shift modes. This is a function that enables you to “double up” the number of commands that you can assign to each button. Just like, for example, the period (full stop) key on your keyboard can be made to print a “>” symbol when you hold down shift, you can assign a second command to each of the buttons on your controller.

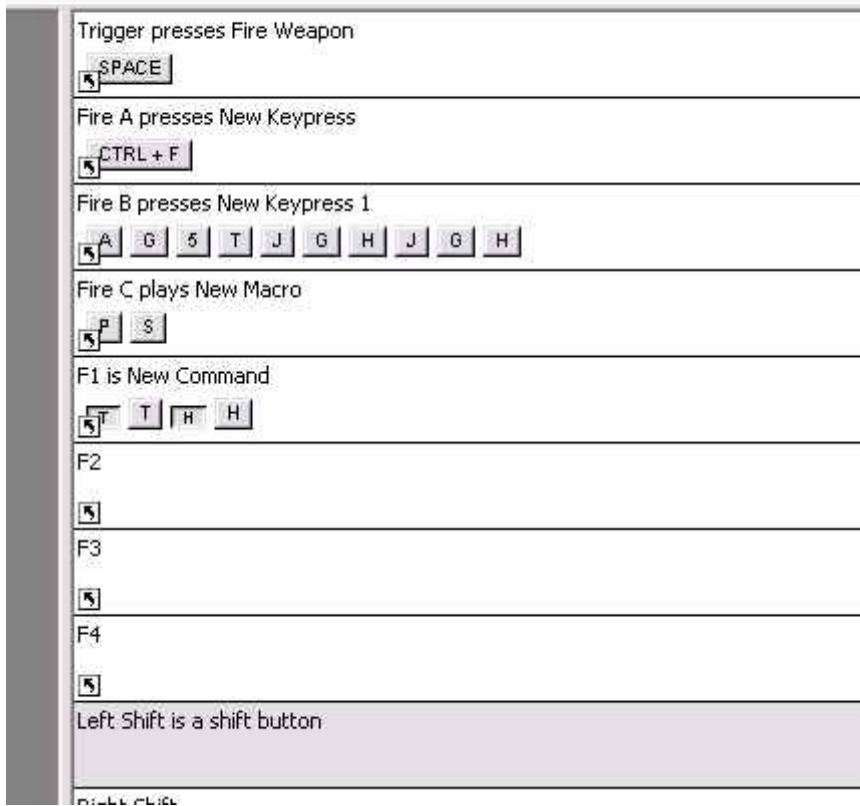
On the Cyborg Stick there are two shift buttons – the ones on the base of the stick with a double arrow symbol on them. On the X36 and X45 the shift button is the pinkie switch that your little finger rests on when gripping the stick. The process for enabling them to act as a shift button rather than a normal button is the same. Using the right click function again, point to the line referring to the shift button and right click.



You can see that there is an option at the top that isn't present on other buttons – the Shift Button option. If you click that then you can see that the programming software now reports that the shift button is acting as a shift button.



From here you have two choices of method to assign commands to your buttons in a shift state. At the top of the Profile Editor window you will see that there is a white bar that says Shift state next to it. In the bar should be the word, Unshifted. If you click that bar you can see that there are two other options – Left Shift and Right Shift. Click the **Left Shift** option and you will now be looking at the list of commands for the buttons when you have the left shift button held down.



You can see that the commands we created in the unshifted mode are carried over into this mode too – the little arrow in the box at the bottom left corner of each command line indicates this. You can now assign the commands you wish to the different buttons, exactly as you have been doing already. You will notice that when you enter a new keystroke onto a button the little arrow in the box disappears indicating that this command only happens in this shift mode and isn't carried over from the unshifted mode. In the picture below you can see that we have put a B keystroke onto the trigger and that arrow is no longer there. Now when we press the shift button on the stick when in the game, the trigger will give us a B keystroke rather than the space bar command that it does in the unshifted mode.















































You can also assign a third set of commands to each button if you enable the right shift button as a shift button.

The second way to input commands into a different shift mode is to switch the view of the Profile Editor to the Data View. The default mode with the 3D model of your controller on the left is called the 3D mode. To switch modes you simply press the Data View button at the top of the Profile Editor, as indicated in the picture below, and you will see the same view as in the picture.

Unshifted	Left Shift	Right Shift
Trigger presses Fire Weapon SPACE	Trigger presses New Keypress 2 B	Trigger presses Fire Weapon [Key]
Fire A presses New Keypress CTRL + F	Fire A presses New Keypress [Key]	Fire A presses New Keypress [Key]
Fire B presses New Keypress 1 A G 6 T J G H	Fire B presses New Keypress 1 [Key]	Fire B presses New Keypress 1 [Key]
Fire C plays New Macro P S	Fire C plays New Macro [Key]	Fire C plays New Macro [Key]
F1 is New Command T T H H	F1 is New Command [Key]	F1 is New Command [Key]
F2	F2 [Key]	F2 [Key]
F3	F3 [Key]	F3 [Key]
F4	F4 [Key]	F4 [Key]
Left Shift is a shift button	Left Shift is a shift button	Left Shift is a shift button
Right Shift	Right Shift [Key]	Right Shift [Key]

This enables you to see what commands you have assigned to each button in all modes. You can also input keystrokes, macros, etc. in this view, exactly as you did in the 3D view.

For the X36 and X45 users, it should be noted here that whilst these controllers have only one shift button you do have the 3 different modes made available to you by the mode switch on the throttle. These are programmed in the exact same way as the Cyborg – in fact it's probably best to use the Data View for these two controllers because, as you can see, it's far easier to keep track of which commands you have programmed in.

Mode 1	Mode 2	Mode 3	Mode 1 + Pinkie
Trigger 	Trigger 	Trigger 	Trigger 
Launch 	Launch 	Launch 	Launch 
Fire A 	Fire A 	Fire A 	Fire A 
Fire B 	Fire B 	Fire B 	Fire B 
Fire C 	Fire C 	Fire C 	Fire C 
Pinkie Switch 	Pinkie Switch 	Pinkie Switch 	Pinkie Switch 
Fire D 	Fire D 	Fire D 	Fire D 
Mouse Fire 	Mouse Fire 	Mouse Fire 	Mouse Fire 
Aux 0 	Aux 0 	Aux 0 	Aux 0 
Aux 1 	Aux 1 	Aux 1 	Aux 1 
Aux 2 	Aux 2 	Aux 2 	Aux 2 

## Saving the Profile

Once you have finished assigning all the commands you want then you must save the profile so that you can access it again later. Simply left click **File** at the top of the Profile Editor...



and then left click **Save**. Just as when saving a document in your word processor your computer will then ask you where you wish to save the profile and what you wish to call it. The name of the profile should ideally be the name of the game that you are making this profile for and once you have done that left click **Save** to save the profile.

For those of you with multiple Saitek controllers you can create folders within the SD02>Software directory that are named after your controllers. For example you could create a folder called X45 and one called Cyborg and save the appropriate profiles in each folder. This then will change the look of the pop-up window in your taskbar next to the clock – you can see what this looks like in the next section which deals with Activating the Profile.

## Activating the Profile

Activating the profile couldn't be easier. Simply right click on the little joystick icon next to your clock and you will now notice that the profile that you saved is at the top of the pop-up menu. In the example below you can see that we called our profile "Test".



If you then left click your profile you will notice that the little joystick icon now has a green square behind it, indicating that a profile has been loaded into the

controller. 

If you wish to clear the profile from your controller at any point then simply right click the joystick icon and click **Clear Profile** – the green circle will disappear indicating that the controller is now cleared of any commands.

Incidentally, if you hold down the Ctrl key on the keyboard and then click the profile in this pop-up menu, instead of configuring the controller with the profile it will launch the profile editor and open the profile for you to edit it. This is obviously a quicker way of opening a profile to edit it than having to run the Profile Editor manually and then opening the profile from in there.

As mentioned in the section above, if you create folders for each controller and store your profiles separately then the pop-up window may look something like this.



You can experiment with this as you see fit but as you can see, it helps when organising your profiles.

## Testing Your Profile

Once you have activated the profile you can obviously test it. A good way to do this is to open up WordPad in Windows – when you then press the buttons on your controller the assigned keystrokes will appear in WordPad. Note, that this will only work with basic keys that you would normally type into a word processor, i.e., letters, numbers and punctuation. The function keys (F1, F2 etc.) shift, alt and ctrl keys obviously won't print anything in a word processor either. We will be including a profile tester in a later release of the programming software so that you can test ALL keystrokes that you input into your profiles.



That concludes the manual for the current version of our SST software. If you have any queries then please [contact your local support representative](#) or feel free to discuss it on our [online forum](#).