

CocoMUD client - Macros - # 2

Macros

CocoMUD supports a system of macros as most MUD clients do, allowing to link a shortcut with an action (as a command to be sent to the server). This document explains the creation of simple and more complex macros.

Basic feature

In its most simple version, a macro is a link between a keyboard shortcut and a command: for instance, if you press Ctrl + F1, CocoMUD may send the "look" command to the server. In this documentation, we will see how to create a macro to link the F1 key with the "north" command, as some MUD clients are already configured to doing.

Adding a macro

There are two ways to add a macro, and as the interface is the easiest alternative, we will see it first.

Through the interface

In the menu bar, select **Game -> Macro**. A dialog box should open with your list of current macros. More likely than not, this list will be empty at first. Click on the **Add** button.

Another dialog opens. The cursor should be on a text entry, but you are not expected to type in it, just press the key combination. For instance, in our example, press the F1 key.

When you do, "F1" should appear in the text entry. You can try to press a lot of different keys. For instance, press the Ctrl key, hold it down and press the I key, then release both keys. You should see in the text area:

```
Ctrl + I
```

You can, in theory, use Shift, Ctrl or Alt in your macros. It's not recommended to use Alt, however, since it can be linked to menu shortcuts (like Alt + F to open the **file** menu). There's not many limitations on what to use for macros. For instance:

```
Shift + Backspace  
Ctrl + F12  
Ctrl + Shift + O  
Ctrl + PavNum8
```

If you wish to connect a macro to a key from the numpad, you should switch it on before pressing the **Add** button, or at least, move in another area before doing so. Otherwise, CocoMUD will associate a macro with the Numpad Lock key, which isn't often desirable.

When you have pressed the desired shortcut to associate with this macro, press Tab to go to the next area. It's also a text area, this time for the command to be sent to the server.

So, following our example, you pressed the **Add** button, then F1, then Tab, you can now type "north".

If you tab again, you find a list of actions. We will study it later, it's only necessary for more complex macros, so you can press tab again until you find the **OK** button. Click on it to add the macro. You should see it in the list of macros.

If you want to change the command to send to the server for a specific macro, you need to edit it (selecting it in the list of macros, and clicking the **Edit** button). However, if you wish to change the shortcut associated with this macro, you can just select it, then press Tab. The cursor will be in another text area, or rather, in a shortcut area, when you can press a new key to associate with this macro. Simply do so, then navigate to **OK** to save the modifications. If you close this dialog box without pressing **OK**, the modifications you have done (including the macros you have added) will be lost.

Back in the client's main window, press the F1 key. The command is sent (silently) to the server and, if you are connected to a server that understands this command, you should receive an answer. In my case, I get:

```
You cannot go in that direction.
```

Through SharpScript

You can also add a macro through SharpScript. This is the way your macros added through the interface are stored, incidentally. A

file, named "config.set", is created in your world settings, containing your macros, aliases and triggers, as well as additional configuration.

To create a macro, you should use the #macro action. It takes two arguments:

- The shortcut to associate with this macro.
- The command to send to the server.

In our example, we could have created our macro using the following SharpScript instruction:

```
#macro F1 north
```

You can actually paste this instruction right in your client's main window and press RETURN, just as if to send a command. Cocomud sees that the command begins with a single sharp symbol and sends it to the SharpScript interpreter. If you add a macro in this way (assuming you made no mistake), it will be visible instantly in the **Game -> Macro** menu.

If one of the parameter is too long, don't forget to place it between braces:

```
#macro {Ctrl + Shift + O} {say that's a rather long line of text.}
```

Braces are necessary when your parameter contains spaces. You can put them even if your parameter doesn't contain spaces, the SharpScript engine will not mind.

Editing a macro

We saw how to edit a macro in the previous section. There's nothing really complicated about it, actually. Just remember that, if you only wish to change a shortcut associated with a macro, you can simply select it in the list of macros (in the **Game -> Macro** menu) and press Tab, then press your new combination of keys to associate with this macro. If you wish to change the text to send to the server, however, you should select the macro and press the **Edit** button. The dialog is the same as described above to add a new macro.

Removing a macro

You can easily remove a macro. In the interface (menu bar -> **Game -> Macro**), select the macro you wish to remove and click the **Remove** button. You will be asked for confirmation. Don't forget to exit the dialog box by clicking on **OK**, otherwise the macro won't be removed.

More complex macros

Macros often connect a keyboard shortcut to a command being sent to the server. But they can actually connect a keyboard shortcut with more complex actions, like playing a sound, displaying a message or a variable, displaying a channel or more. The interest is not always obvious at first glance. The example of more complex macros can be found in some configurations, when the world provides audio prompting, for instance.

The general idea of audio prompting is that the line of prompt (where you can find health points and movement points and that appears at each command, or almost) is hidden. It is captured by a trigger. The information is stored in variables. That's beyond the scope of this document to explain how, but the important point is, you can configure a macro to display this variable. For instance, if you press Ctrl + H, you should get the health points (captured by the trigger).

We have used the second text area to send commands to the server. The truth is, you can also type SharpScript instructions in this area. But as writing SharpScript can be a bit technical, you can also use the SharpScript editor. It is not as complicated as it sounds: when you add or edit a macro, simply leave the second text area blank. Tab to the list of actions and select one. For instance, select "Display a message and send it to the screen reader". Click on the **Add action** button. You will be prompted for additional information (in this case, what message to display and how to display it). When you're done, click **OK**. In the second text area will be your SharpScript instruction. You don't need it, it's just a landmark. Click on the **OK** button several times to exit the dialog by saving your modifications.