Aliases in CocoMUD

Aliases are a special feature of MUD clients that can be used to shorten commands. An alias is itself a command, but instead of being sent to the server as is, it is first analyzed by the client, modified and extended if necessary. Some aliases don't leave the client at all.

This document will introduce the concept of aliases and explain how to create and manage them.

Adding an alias

You can create an alias using the interface or through the SharpScript syntax. The interface being easier at first glance, it will be explained in first.

Through the dialog box

In the menu bar, select Game -> Aliases. You will find yourself in a dialog listing the current aliases for this world. Note that aliases are usually created in a world, being shared by characters. This will change in future versions.

In this dialog, you can add, edit or remove aliases. By default, the list of alias will most likely be empty, so you can select **Add** to add one.

You will then have to choose an alias name. This is the command you want to create, basically. In this example, we're going to create a "tts" alias that enables/disables the text-to-speech in the client.

Enter "tts" in this field. When you press Tab, you will be presented with a list of actions to be performed when you type this command. For instance, you may want the "rs" alias to send two commands at the MUD server: "reload" and "shoot". In this case, you would select the action named "send one or more commands to the server".

In our example, we'll select "Enable/Disable TTS", since that's what we want to do when we type "tts" in the client. Tab again and click the **Add action** button.

You will be prompted with a dialog asking you to review parameters to this action. For this action, there is none, so you can press "OK" to add the action.

The new action has been added. The cursor will be put in the list of current actions linked to this alias. You can link an alias with several actions (even with the same action several times, to play several sounds, for instance, although it might not be very useful).

The list of current actions may be a bit scary. It contains summarized information. If you have selected the "Enable/Disable TTS", it will create the action #tts. That's a shortcut, a notation used by the SharpScript engine, and you don't need to worry much about it, unless you want to play with the scripting system in CocoMUD.

To summarize the dialog when you add or edit an alias, you will find:

- The alias' name (a text edit).
- A list of actions that are currently linked with this alias. If the alias isn't linked with any action, the list won't appear.
- A button to edit the selected line of action. If the alias isn't linked with any action, the button won't appear.
- A button to remove this line of action from the alias. If the alias isn't linked with any action, the button won't appear.
- A list of actions that can be linked to this alias.
- The button to add this action.

You should get accustomed to this dialog, as it will be present for most configuration involving SharpScript. It is, in fact, the SharpScript editor that do not ask you to edit the configuration file by hand. You will find more examples in the rest of the documentation.

To save the alias, don't forget to select "OK" several times, until the dialog box is closed.

If you then type "tts" in the client, you should see (and hear):

```
TTS off.
```

Type it again to switch it on. The text-to-speech will be disabled or enabled using this alias.

SharpScript syntax

You can also add an alias by editing your configuration file and adding it there. This solution might be preferred by some.

Open your configuration file for this world. You will find it in the "worlds" directory. Select the directory containing your world, then the "config.set" file. You can open it with a small editor like notepad.

To add an alias, use the #alias action with two parameters:

- The alias' name.
- The alias' actions to be performed.

If you want to create a "tts" alias that will enable or disable the text-to-speech, add in your file the following line:

```
#alias tts #tts
```

If you have several actions, don't hesitate to describe the alias on several lines:

```
#alias go {
    #play sounds/go.wav
    #say {Here we go!}
}
```

For an explanation of the SharpScript syntax, refer to the section describing SharpScript.

Editing an alias

At any time, whether you have created the alias in the dialog box or in the configuration file, you can edit it. If you have created it in the configuration file, you can modify it through the interface, and vice versa.

Remember that the configuration is loaded when you select the world, which will most likely happen when you open the software. If you modify the configuration file, restart the software to take it into account.

Removing an alias

You can remove an alias through the dialog box or by editing the configuration file. If you do it through the dialog box, remember the changes will be lost if you close the dialog but do not use the "OK" button.

Aliases with variables

CocoMUD provides a system of variables. They can be used anywhere in your script or in any feature using the SharpScript engine (aliases, macros, triggers...).

Variables can be more than useful for aliases to create shortcut commands with unknown parameters. For instance:

> I want to create an alias that begins by =. Everything after the = sign should be sent to the server using the say command. If I type =hello ! in the client, then it should send say hello ! to the MUD.

For this case study, we will use variables. The alias begins by an equal sign (=), but we don't know what's after that. How to handle that context? We will use the * sign in our alias name, which means "about everything, as long as necessary".

So if we enter =* as an alias name, CocoMUD will understand it as meaning "everything starting with an equal sign".

But we don't only need for the alias to be activated. We need for it to send a command with what has been entered after the equal sign. Here is the process step by step, explained below:

- In the menu bar, select **game** then **aliases**.
- Click the **add** button to add a new alias.
- In the alias field, enter =* (an equal sign and the asterisk sign).
- Tab to move to the list of actions that can be linked to this alias.
- Select "send one ore more commands to the server", tab and click the **add action** button.
- You'll be asked to configure this action. Tab until you hear "Commands to be sent to the server". In this field, type say $1 (the $1 will be explained below).
- Select **OK**. The action has been linked, and you should see it in this field:

```
    #send {say $1}
    ```

    What it means will be detailed below.
- Tab until you find the **OK** button and click on it. You will find yourself in the list of aliases, press **OK** again.

If you type in the client:

```
=hello !
```

The alias will send "say hello !" to the server.

What happens is pretty simple:

- You enter "=something".
- CocoMUD identifies a matching alias.
- The "something" part (after the equal sign) will be captured and put in a variable, called $1.
- When you send "say $1" to the server, CocoMUD replaces it $1 with the "something" part.

The variable is called $1 because you can create a lot of variables. If you enter the alias:

```
-*+*
```

You will be able to send:

```
remove $1
wear $2
```

And then type:

```
-cloak+armor
```

Variables in SharpScript are described more completely in the section describing SharpScript.