

# **M.S.C.I.**

**aka Script Editor**

**Documentation**

**UNDER CONSTRUCTION!**

**© 764 by Bala Gi A.I. Software Inc.,  
Kingdom End, X-Universe**

## **Index**

- **Purpose of this document**
- **What is the M.S.C.I.?**
- **What do I need to write scripts?**
- **The Basics**
  - **Interface and Controls**
  - **Writing a small script**
  - **Commonly used commands**
- **Advanced**
  - **More commands**
  - **Script debugging**
  - **Embedding scripts into the command console**

# Purpose of this document

This document is intended to give you an overview of how to use the MSCI and what can be done with it.

However this can by no means considered a complete manual as the true potential of the Script Editor is far beyond the scope of this document.

## What is the M.S.C.I.?

The **M.S.C.I.** is the „Manual Ship Computer Interface“ also known as Script Editor.

Here is an excerpt from the brochure of the MSCI:

---

Ever wanted to have better control over your ship?

Ever wanted to manage and control your property with the latest technology?

Ever wanted to put your own implementations of tactics and artificial intelligence into your ship?

If yes, you have come to the right place! With **M.S.C.I.** you can modify the behaviour of your ships down to the finest detail.

- Equip your ships with different default behaviours, which are individually modifiable for every shiptype!
  - **Configure your own ship-type-dependent comm menu to assign your remote commands to your ships fast and efficiently.**
  - It is also possible to give your ship an autopilot command while you jump into the turret to bug some split.
  - Control the turrets of your destroyer and give them different tactics.
  - Create commands to assign routes to your ships or increase their fighting behaviour.
-

The purpose of the MSCI was to allow easy implementation of new and partly automated AI behaviour. However during development it became obvious that the MSCI was capable of doing much more than just allowing simple „go here, do this, go there, do that...” and so it evolved. Now the MSCI is a very powerful program that would be better described as a modding tool rather than an editor.

But with power comes danger.

Please keep in mind that the MSCI has a great influence on the game. It is not rare that a script can cause the game to crash if it is programmed badly. (This is the voice of experience speaking ☺)  
Once you had a look at the commands you will notice that the MSCI features some methods that not only affect the behaviour of objects but also their very existence.  
This is the point where scripting turns into modding and not only the AI is influenced but your whole gaming experience – whether it is a good or a bad influence is of course up to you and the scripts you use or write.

## What do I need to write scripts?

First of all you will need the game to write and use scripts. Currently there is no external editor for script writing available.

The X<sup>2</sup> Scripting language is effectively an advanced programming tool. While I do not wish to discourage those who feel curious as to its usage, I must warn you that it should be attempted only by those who have a clear idea of the basic principles of programming, or have prior experience in programming or scripting languages such as C or JavaScript...

Does that mean you have to learn a programming language?

No – the MSCI is quite straight forward and you don't have to remember complicated commands as it's all available from different menus. Mostly you can figure out what those commands/operators do either from their name or with some logical thinking ;)   
Still it's a fact that any kind of programming knowledge helps a lot and will speed up this process extremely.

I suggest to print this document for quick reference. **Green** words are explained in the appendix.

# The Basics

One of the most frequently asked questions is „Where can I find it?“ When you start a game the MSCI is a „hidden“ feature that needs to be enabled using a cheatcode.

While in space or inside a station (but not connected to the trade network) type:

**Thereshallbewings** (note the captial T)

You should hear a confirmation-beep. (No menu should open, if it did not work try again and type slower) From now on the Command Console screen features another item, the Script Editor.



This editor can be accessed from different menus:

**Main-Menu -> Command Console -> Script Editor**

or

**Ship Menu -> Command Console -> Script Editor**

are the most common methods.

# Interface and Controls

When you launch the script editor you will see a screen with a warning message:

**The Script-Editor is a very powerful tool which can have big influence on parts of the game. Egosoft cannot take responsibility for any damage or weird behaviour caused by using selfmade scripts.**

<l>	load script from script-dir
<c>	copy script
<r>	execute script
<del> script	delete
<return>	edit script
<b>Actions while editing a script:</b>	
<up/down>	navigate through lines
<left/right>	navigate through expressions
<return>	functions
menu	
<ins>	insert
new line	
<del>	delete line
<c>	copy line
<x>	cut line
<v>	paste line
<.>	comment out line
<,>	uncomment line
<home>	insert
expression operand	
<end>	remove
expression operand	
<esc>	close
script and save	

Now select **OK** and you see the MSCI main menu.



From this menu you can access the previous screen by selecting **Help**. (Although it might be easier to just print the page...)

Now let's get to the real thing, select **Script Editor** and press **return**. I will come back to this screen later on.

The screen you see now is referred to as the „Script Browser“. From here you can access all existing un-encrypted scripts as well as create new ones. (The scripts that ship with the game are encrypted in .pck format and do not show up in this list. User-created scripts are not encrypted.)

Feel free to take a look at the existing scripts. But I suggest (unless you know exactly what you are doing) not to change or at least not to save a modified version of these scripts.

From personal experience I can tell you that „learning by doing“ works pretty well for the Script Editor so the following sections will guide you through the creation of the simplest of all scripts and one basic trading script that will get one of your ships to trade a certain ware.



Use the arrow keys, your mousewheel or even the coolie hat to scroll to **<New Line>** press return and the „Select Script Command“-screen shows up.



#### »» **General Commands**

This section houses basic programming commands for conditions, loops, arrays or script calls

#### »» **Audio Commands**

As the name suggests here you can find commands to play certain audio samples or send incoming messages

#### »» **Logbook Commands**

Here you can find commands to output data into the logbook of an object or even the player

#### »» **Fly Commands**

Guess what – lots of commands for space flight, evasive action, move to position, follow... it's all there

#### »» **Trade Commands**

Find a station with good offers, buy, sell, get free cargo space, get volume of ware, get product ware...

#### »» **General Object Commands**

Get owner, get shield strength, get homebase – find out everything you need to know about an object (ships, stations...)

#### »» **Universe and Sector Commands**

Everything you need for navigation on a galactic level, get sectors by coordinates, specific gates, or measure the universe

# Hello World!

If you ever read other documentations that had to do with programming or scripting you probably noticed that many of them use a "Hello World!" output as an example. I am going to continue that tradition.

The easiest way to output data in a readable form is using the logbook of the player.

The command for writing to the player-log only accepts a single **variable/string/number** so if you want to output a text that has been assembled from different parts you need to define a variable that stores these text parts and output that variable.

Let's begin:

-> **General Commands** -> **<RetVal/If><Expression>** -> chose **variable** -> type "output" -> return twice -> chose **string** -> type "Hello World!" -> return

You should see this:

```
001 $output = Hello World!
```

Now we want to write this text to the playerlog.

-> **Logbook Commands** -> **write to player logbook <Value>** -> return -> chose \$output

```
001 $output = "Hello World!"
002 write to player logbook $output
```

Save and close the script.

Now in order to run the script highlight it in the Script Browser and press **<r>** (see list of key commands in „**Interface and controls**“). You are prompted to select an object to run this script on. This script does not depend on a specific object in the game and only runs once so it does not matter whether you select **null** or a **ship/station** here.

## **(!) Warning (!)**

**Do not run scripts that are supposed to run continuously on "null". As of yet there is no way to stop a running script that is not object bound. A buggy script can cause serious, unwanted sideeffects such as savegame corruption and game crashes!**

Now check your logbook, you should see a new entry: "Hello World!"  
Congratulations :)

This method is rather silent and if you want to make the player aware of the new message you either have to play a soundfile or display the message right away. (The latter is not very nice though)

For an audio message go to -> Audio Commands -> send incoming message to player (you could also chose to play a specific soundfile, for a list of available soundfiles see appendix)  
The second option, while working fine, is not recommended.

# Writing a useful small script

In general before starting to write a script you should have an idea of what your script should do, how it should be done and what you need to do for it.

Concept idea:

- Freightship should continuously buy/sell a certain ware for best profit
- Ware is specified by player upon start of the script
- Player specifies how far the ship is allowed to fly

This should do for now.

Select **<New Script>** in the Script Browser and chose a name for the new script. I will use „**tutorial.tradeware**“.

You can specify the Version of your script, add a small Description and a Command Name. For now leave these options as is.

Since we plan to let the player chose what ware should be traded we need to pass this ware onto the script using an argument variable.

Select **<New Argument>** and press return.

Creating an argument consists of 3 steps:

- > specify the name for the variable
- > specify the datatype for the variable
- > specify a „helptext“ to inform the player what he has to enter/select.

You are now prompted to enter a name for the variable that will be used in the script. Type „**ware**“ and confirm with return.

Next you will be shown a list of datatypes. Among those are „**Number**“, „**String**“, „**Value**“ and further down the list (near the bottom)

„**Var/Ware**“ select the latter. Last but not least type your helptext. I wrote „Chose ware to trade with.“.

Argument 2 will be the number of jumps the ship is allowed to make.

Repeat the steps from above for the argument name „**max.jumps**“ and as datatype chose „**Number**“.

Since we want the script to run continuously we need to define a loop. Go to **General Commands** and select **<RetVar/IF><Expression>**. In the next menu scroll down and select **while**