

# **Module**

Paul Manias

**COLLABORATORS**

	<i>TITLE :</i> Module		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Paul Manias	July 26, 2024	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Module</b>	<b>1</b>
1.1	Object: Module . . . . .	1
1.2	Object: Module . . . . .	1
1.3	Object: Module . . . . .	2
1.4	Object: Module . . . . .	2
1.5	Object: Module . . . . .	2
1.6	Object: Module . . . . .	3
1.7	Object: Module . . . . .	3
1.8	Module: Init() . . . . .	3

---

# Chapter 1

## Module

### 1.1 Object: Module

#### OBJECT DOCUMENTATION

Name: MODULE AUTODOC  
Version: 0.9 Beta  
Date: December 1997  
Author: Paul Manias  
Copyright: DreamWorld Productions, 1996-1997. All rights reserved.

#### CHANGES VERSION 0.9B

Added: Module->Name

### 1.2 Object: Module

#### OBJECT

Name: Module  
Module: Kernel  
Version: 1  
Type: Complex

#### DESCRIPTION

The module object serves as a launching pad for all module files. If you want to write a new module for public use, read the relevant section in the general documentation.

#### ACTIONS

The Module object supports the following actions:

- Free ()
- Get ()
- \* Init ()

#### STRUCTURE

The Module object consists of the following public fields:

- Name Name of the module.
- Number ID of the associated module, if any.

Functions Function jump table.  
Version Version number.  
Revision Revision number.

### 1.3 Object: Module

#### FIELD

Name: Name  
Type: BYTE \*  
On Change: Cannot change after initialisation.  
Status: Read/IWrite

#### DESCRIPTION

This field points to a string that specifies the name of the module. This name will be used to load the module from the system directory, so it actually reflects the filename of the module. It is also possible to specify sub-directories before the module name itself.

This field is optional, but if you do not know the ID of a module, you will have to give a Name so that the module can be found.

### 1.4 Object: Module

#### FIELD

Name: Functions  
Type: APTR  
On Change: Cannot change after initialisation.  
Status: Read/IWrite

#### DESCRIPTION

### 1.5 Object: Module

#### FIELD

Name: Number  
Type: WORD  
On Change: Cannot change after initialisation.  
Status: Read/IWrite

#### DESCRIPTION

This field identifies the number of the Module. All modules registered with DreamWorld Productions will have an ID and this is the fastest way to reference them.

---

## 1.6 Object: Module

### FIELD

Name: Revision  
Type: LONG  
Inheritance: Init()  
On Change: Cannot change after initialisation.  
Status: Read/IWrite

### DESCRIPTION

When opening the Module, the value that you insert here will specify the minimum revision of the module that you require. If the module passes the revision criteria, Init() will update this field to reflect the actual revision of the initialised module.

Note that the revision is closely tied to the Version number. So, when asking for version 1 revision 4 (1.4), when the available version is 2.2, you will still open the module successfully even though you have specified a "lower" revision number.

### SEE ALSO

Field: Version

## 1.7 Object: Module

### FIELD

Name: Version  
Type: LONG  
Inheritance: Init()  
On Change: Cannot change after initialisation.  
Status: Read/IWrite

### DESCRIPTION

When opening the Module, the value that you insert here will specify the minimum version of the module that you require. If the module passes the version criteria, Init() will update this field to reflect the actual version of the initialised module.

### SEE ALSO

Field: Revision

## 1.8 Module: Init()

### ACTION

Name: Init()  
Object: Module  
Short: Load in a system module and initialise it for function calls.

### DESCRIPTION

Loads in a module and initialises it ready for function calls. After successful initialisation you will be able to interface with the module.

---

SEE ALSO

Kernel: Init()