

# **Keyboard**

Paul Manias

|                      |
|----------------------|
| <b>COLLABORATORS</b> |
|----------------------|

|               |                            |              |                  |
|---------------|----------------------------|--------------|------------------|
|               | <i>TITLE :</i><br>Keyboard |              |                  |
| <i>ACTION</i> | <i>NAME</i>                | <i>DATE</i>  | <i>SIGNATURE</i> |
| WRITTEN BY    | Paul Manias                | May 28, 2025 |                  |

|                         |
|-------------------------|
| <b>REVISION HISTORY</b> |
|-------------------------|

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
|        |      |             |      |

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Keyboard</b>                        | <b>1</b> |
| 1.1      | Keyboard Object Documenation . . . . . | 1        |
| 1.2      | Object: Keyboard . . . . .             | 1        |
| 1.3      | Object: Keyboard . . . . .             | 2        |
| 1.4      | Object: Keyboard . . . . .             | 2        |
| 1.5      | Object: Keyboard . . . . .             | 2        |

# Chapter 1

# Keyboard

## 1.1 Keyboard Object Documentation

### OBJECT DOCUMENTATION

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

## 1.2 Object: Keyboard

### OBJECT

Name: Keyboard  
Module: Keyboard  
Version: 1  
Type: Simple

### DESCRIPTION

The purpose of the Keyboard object is to provide a system interface between applications and the computer keyboard. The object bases itself around a system buffer that contains all keyboard inputs from the user. Updates to the system buffer are maintained by a hardware interrupt that reads key-codes and outputs their ANSI representations.

Currently you can read the key-codes using the Query() action. The keyboard object will soon allow you to Write() codes to the system buffer, allowing you to send codes to all other programs in the system.

### ACTIONS

The Keyboard object supports the following actions:

- Free() Free the object.
- Get() Get a new keyboard structure.
- Init() Initialise a keyboard object.
- Query() Read all keys since last.

### STRUCTURE

---

The Keyboard object consists of the following public fields:

Size      The size of the buffer in bytes.  
Buffer    Pointer to your personal key buffer.  
AmtRead   Current amount of keys read.

### 1.3 Object: Keyboard

#### FIELD

Name:        Size  
Type:        LONG  
Default:     512 bytes.  
On Change:   Cannot change after initialisation.  
Status:      Read/IWrite

#### DESCRIPTION

This field defines the byte size of your key buffer. To illustrate, a setting of 256 would allow your Keyboard object to store a maximum of 256 key strokes.

#### SEE ALSO

Field: Buffer

### 1.4 Object: Keyboard

#### FIELDS

Names:       Buffer  
Type:        BYTE \*  
Inheritance: Allocated on initialisation.  
Status:      Read Only.

#### DESCRIPTION

This field points to the buffer that contains the keystrokes for your Keyboard object. Its length matches that of the Keyboard->Size field.

#### SEE ALSO

Field: Size

### 1.5 Object: Keyboard

#### FIELD

Name:    AmtRead  
Type:    WORD  
Status:   Read Only.

#### DESCRIPTION

Whenever you call the Query() function, this field will be updated to tell you the amount of keystrokes that were read into the Buffer. If no keys were pressed since the last time you called Query(), AmtRead will read as NULL.

---

## NOTE

The amount of keys read will never exceed the number specified in the Size field.

## SEE ALSO

Field: Buffer