

# Ini File Object

## Version 0.1

**Richard D. Clark**

The ini file object creates and manages ini files for your FreeBasic program. To use the object add **#Include Once "inifile.bi"** to your project. The object uses the StringList object to manage the ini file contents. The ini file object only reads and writes string values, so any numeric values will need to be converted to strings before saving in the ini file.

The section and key names are case insensitive.

The ini file object manipulates the ini file in memory to avoid a disk write for each value. To update the ini file on disk, be sure to call the UpdateFile function to save the changes to the disk file.

### Ini File Format

```
[SECTION 1]
ITEM2=This is another string.
ITEM1=This is a new string.
[SECTION 2]
ITEM1=Yet another string.
```

The section names have the [ ] characters around them in the file, but do not use the [ ] characters when accessing a section. That is, [SECTION 1] would be called as "Section 1". The section keys are the values before the = character, and the key data are the values after the = character. To access a key value you need to pass both the section name and key name. For example, to return "This is a new string." you would call ReadString with "Section 1" for the section and "Item1" for the key.

### Required Files

inifile.bi  
stringlist.bi

## Interface

### Constants

```
#Define FALSE 0
#Define TRUE (Not FALSE)
```

### Declare Constructor (inifile As String)

Object constructor. Pass a file name to the constructor when creating the object. If the file exists, the

file is loaded into the internal StringList.

Ex:

```
Dim myIni As iniobj = iniobj("test.ini")
```

### **Declare Property Version () As String**

Returns the current version of the iniobject as a string.

Ex:

```
Print myIni.Version
```

### **Declare Property InitStatus () As Integer**

Returns TRUE if the object was initialized properly, FALSE if not.

Ex:

```
If myIni.InitStatus = FALSE Then
    Print "Error initializing ini object."
    Sleep
End
EndIf
```

### **Declare Property IniFileName () As String**

Returns the ini file name as a string.

Ex:

```
myFile = myIni.IniFileName
```

### **Declare Property IniFileExists () As Integer**

Returns TRUE if ini file exists or FALSE if not.

Ex:

```
ret = myIni.IniFileExists
```

### **Declare Property Count() As Integer**

Returns the number of lines in the ini file.

Ex:

```
cnt = myIni.Count
```

### **Declare Function Strings(index As Integer) As String**

Returns the string located at Index in the internal StringList.

Ex:

```
Print myList.Strings(index)
```

### **Declare Function SectionExists (section As String, ByRef index As Integer) As Integer s**

Returns TRUE if section name exists, False if not. Section is passed without the [ ] characters. The

index of the section is returned in the Index parameter if found.

Ex:

```
If myIni.SectionExists("section 1", index) = TRUE Then
    Print myIni.Strings(index)
EndIf
```

### **Declare Function KeyExists (section As String, skey As String, ByRef index As Integer) As Integer**

Returns TRUE if key exists in section, FALSE is not. Section is passed without the [ ] characters. Skey is the value before the = sign in the ini file. If the key is found, the index of the key is returned in index.

Ex:

```
If myIni.KeyExists("section 1", "item1", index) = TRUE Then
    Print myIni.Strings(index)
EndIf
```

### **Declare Function ReadString (section As String, skey As String, default As String) As String**

Returns the string value in section and located at skey. This is the value after the =. If section, key or value is not found, the default value is returned.

Ex:

```
sret = myIni.ReadString("section 2", "item1", "No Value")
Print "Return String = "; sret
```

### **Declare function UpdateFile() As Integer**

Updates the disk file with the current ini file data. Returns TRUE if successful, FALSE if an error occurs. All changes to the ini file occur in memory and are not written to disk until the UpdateFile function is called.

Ex:

```
ret = myIni.UpdateFile()
If ret = FALSE Then
    Print "Could not update file."
Else
    Print "Updated file."
EndIf
```

### **Declare function DeleteSection(section As String) As Integer**

Deletes section and all key/values associated with section. Returns TRUE if successful, FALSE if an error occurs.

Ex:

```
ret = myIni.DeleteSection("section 2")
If ret = TRUE Then
    Print "Deleted section 2."
EndIf
```

### **Declare function DeleteKey(section As String, skey As String) As Integer**

Deletes key/value in section. Returns TRUE if successful, FALSE if an error occurs.

Ex:

```
ret = myIni.DeleteKey("section 2", "item1")
If ret = TRUE Then
    Print "Deleted key item1 from section 2."
EndIf
```

### **Declare Sub WriteString (section As String, skey As String, value As String)**

Writes the string value within section and at key. If the section or key do not exist, they are created.

Ex:

```
myIni.WriteString "Section 1", "item1", "This is a new string."
```

### **Declare Sub GetSections (ByRef slist As stringlist)**

Returns all the section names in the passed stringlist. If no sections are found, the stringlist count will be 0.

Ex:

```
myIni.GetSections(myList)
For index = 0 To myList.count - 1
    Print myList.Strings(index)
Next
```

### **Declare Sub GetSectionKeys (section As String, ByRef slist As stringlist)**

Returns all the key names within section in the passed string list. If the section is not found, or the section does not contain any key names, the stringlist count will be 0.

Ex:

```
myIni.GetSectionKeys "section 1", myList
For index = 0 To myList.count - 1
    Print myList.Strings(index)
Next
```

### **Declare Sub GetSectionKeyValues (section As String, ByRef slist As stringlist)**

Returns all the key/value, i.e., ITEM1=This is a new string., within section in passed stringlist. If the section is not found, or the section does not contain any key names, the stringlist count will be 0.

Ex:

```
myIni.GetSectionKeyValues "section 1", myList
For index = 0 To myList.count - 1
    Print myList.Strings(index)
Next
```