

# Ini-Datei-Object

Version 0.1d

**Richard Dean Clark**

**Übersetzung: Jörg Zühlke, aka ytwinky**

Das Ini-Datei-Objekt erzeugt und verwaltet Ini-Dateien für Ihr FreeBASIC-Programm.  
Um das Objekt benutzen zu können, muß dem FB-Projekt folgende Anweisung hinzugefügt werden:  
`#include Once "inifile.bi"`

Das Objekt benutzt das StringList-Objekt, um den Ini-Dateiinhalte zu verwalten. Das Ini-Datei-Objekt liest und schreibt nur String-Werte, deshalb müssen alle numerischen Werte in String-Werte umgeformt werden, bevor sie in die Ini-Datei geschrieben werden.

Bei Sektions- und Schlüsselnamen wird Groß-/Kleinschreibung nicht beachtet.

Das Ini-Datei-Objekt bearbeitet die Ini-Datei im Speicher, damit nicht bei jeder Wertänderung in die Datei geschrieben werden muß. Zum Update der Datei auf dem Laufwerk, muß die `UpdateFile`-Funktion aufgerufen werden, damit die Änderungen endgültig gespeichert werden.

## Ini-Datei Format

```
[Sektion 1]
Punkt2=Dies ist noch ein String
Punkt1=Dies ist ein String
[Sektion 2]
Punkt1=Ein ganz anderer String
```

Die Sektionsnamen müssen von `[]` umgeben sein, die aber beim Zugriff nicht mit angegeben werden dürfen. Also würde `[Sektion1]` als "Sektion1" aufgerufen werden.

Die Sektionsschlüssel sind die Werte vor dem '='-Zeichen, die Schlüsselwerte sind die Zeichen nach dem '='-Zeichen. Um auf einen Schlüsselwert zuzugreifen, muß der Sektionsname und der Schlüsselname übergeben werden. Soll z.B. "Dies ist ein String" zurückgegeben werden, muß also "Sektion1" als Sektionsname und "Punkt1" als Schlüsselname übergeben werden.

## Erforderliche Dateien:

inifile.bi  
stringlist.bi

## Interface

### Konstanten

```
#Define False 0
#Define True (Not False)
```

### Declare Constructor (Inifile As String)

Objekt-Constructor. Übergeben Sie dem Constructor einen Dateinamen beim Erstellen des Objekts.

Wenn es diese Datei gibt, wird sie in die interne StringList geladen.

Bsp:

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

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

Gibt die aktuelle Version vom iniobject als String zurück

Bsp:

```
Print myIni.Version
```

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

Gibt TRUE zurück, wenn das iniobject richtig initialisiert wurde, sonst FALSE

Bsp:

```
If myIni.IniStatus=FALSE  
    Print "Fehler beim Initialisieren von iniobj."  
    Sleep  
    End  
End If
```

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

Gibt den Ini-Dateinamen als String zurück

Bsp:

```
IniDatei=myIni.IniFileName
```

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

Gibt TRUE zurück, wenn es die Ini-Datei gibt, sonst FALSE

Bsp:

```
erg=myIni.IniFileExists
```

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

Gibt die Zeilenanzahl der Ini-Datei zurück

Bsp:

```
anz=myIni.Count
```

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

Gibt den String mit Nummer Index aus der IniDatei zurück

Bsp:

```
Print myIni.Strings(Index)
```

### **Declare Function SectionExists (Sektion As String, byRef Index As Integer) As Integer**

Gibt TRUE zurück, wenn es den Sektionsnamen gibt, FALSE, wenn nicht.

Sektionsname wird ohne [] übergeben, der Sektionsindex wird im Index-Parameter zurückgegeben, wenn gefunden.

Bsp:

```
If myIni.SectionExists("Sektion 1", Index) = TRUE Then  
    Print myIni.Strings(Index)  
End If
```

### **Declare Function KeyExists (Section As String, Schluessel As String, ByRef Index As Integer) As Integer**

Gibt TRUE zurück, wenn es den Schlüssel in der Sektion gibt, FALSE wenn nicht.

Sektion wird ohne [ ] Zeichen übergeben. Schluessel ist der Wert vor dem = Zeichen in the Ini-Datei.

Wird der Schlüssel gefunden, wird der Index vom Schlüssel in Index. zurückgegeben.

Bsp:

```
If myIni.KeyExists("Sektion 1", "Punkt1", Index) = TRUE Then  
    Print myIni.Strings(Index)  
End If
```

### **Declare Function ReadString (Sektion As String, Schluessel As String, default As String) As String**

Gibt den String-Wert in Sektion zurück, der zu Schluessel gehört. Dies ist der Wert nach dem =.

Wird Sektion, Schluessel oder der Wert nicht gefunden, wird der Default-Wert zurückgegeben.

Bsp:

```
erg = myIni.ReadString("Sektion 2", "Punkt1", "Kein Wert")
```

```
Print "Rückgabewert = "; erg
```

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

Die Ini-Datei auf dem Laufwerk wird mit den aktuellen Ini-Daten geupdatet.

Gibt bei Erfolg TRUE zurück, FALSE wenn ein Fehler auftritt.

Alle Änderungen an den Ini-Daten geschehen im Speicher und werden nicht auf die Platte geschrieben, bis die

UpdateFile-Funktion aufgerufen wird

Bsp:

```
erg = myIni.UpdateFile()
```

```
Print "Datei-Update war " &*IIf(erg, @"" , @"nicht ") &"erfolgreich"
```

[A.d.Ü.] Da konnte ich nicht widerstehen/Sorry, couldn't resist ^^

### **Declare Function DeleteSection(Sektion As String) As Integer**

Löscht Sektion und alle Schlüssel/Werte, die zu dieser Sektion gehören.

Gibt bei Erfolg TRUE zurück, FALSE wenn ein Fehler auftritt.

Bsp:

```
erg = myIni.DeleteSection("Sektion 2")
```

```
If erg = TRUE Then
```

```
    Print "Sektion 2 gelöscht!"
```

```
End If
```

### **Declare Function DeleteKey(Sektion As String, Schluessel As String) As Integer**

Löscht Schlüssel/Wert in Sektion. Gibt bei Erfolg TRUE zurück, FALSE wenn ein Fehler auftritt.

Bsp:

```
erg = myIni.DeleteKey("Sektion 2", "Punkt1")
```

```
If erg = TRUE Then Print "Schlüssel Punkt1 aus Sektion 2 gelöscht!"
```

### **Declare Sub WriteString (Sektion As String, Schluessel As String, Wert As String)**

Schreibt den String-Wert in Sektion bei Schluessel. Wenn es Sektion oder Schluessel nicht gibt, werden sie erstellt.

Bsp:

```
myIni.WriteString "Sektion 1", "Punkt1", "Dies ist ein neuer String."
```

### **Declare Sub GetSections (ByRef Liste As StringList)**

Gibt alle Sektionsnamen in der übergebenen StringList zurück. Werden keine Sektionen gefunden, ist StringList-Anzahl=0

Bsp:

```
myIni.GetSections(myList)
```

```
For Index = 0 To myList.Count - 1
```

```
    Print myList.Strings(Index)
```

```
Next
```

### **Declare Sub GetSectionKeys (Sektion As String, ByRef Liste As StringList)**

Gibt alle Schlüsselnamen in Sektion der übergebenen Liste zurück. Wenn die Sektion nicht gefunden wrd oder die

Sektion keine Schlüsselnamen enthält, dann ist Count=0.

Bsp:

```
myIni.GetSectionKeys "Sektion 1", myList
```

```
For Index = 0 To myList.Count - 1
```

```
    Print myList.Strings(Index)
```

```
Next
```

```
Declare Sub GetSectionKeyValues (Sektion As String, ByRef Liste As StringList)
```

Rückgabe aller Schlüssel/Werte, z.B., PUNKT1=Dies ist ein neuer String, innerhalb von Sektion in übergebener StringList.

Wird Sektion nicht gefunden oder die Sektion enthält keine Schlüsselnamen, dann ist StringList count=0.

Bsp:

```
myIni.GetSectionKeyValues "Sektion 1", myList
```

```
For Index = 0 To myList.Count - 1
```

```
    Print myList.Strings(Index)
```

```
Next
```