

## Updating an Initialization File for Additional Values – Jim Grossman

September 11, 2009 --

[jbross1](#)

Many programs use an Initialization file to hold program values that are generally changed by the users to suit their needs. Most programmers use a sequential file which can hold any number of values and can easily mix strings and numbers. No matter how hard you try, sooner or later you are going to want to add a new value or two which means updating the \*.ini file. If you are distributing your program to an end user, you can be left in a quandary about how to update the file without the user even noticing that it is being done. I have been using the following method in some of my programs that is transparent to the user.

This method relies on the On Error statement to redirect program flow to a section of code that will repair the ini file with the original values set by the user and adds the new values which would be set to the default settings. It does not rely on an external program that is run by an installer but does its work the first time the new version of your program is run.

The following program will create an initialization file for the purposes of this demonstration and when you click on the Run Program button the program will flow to the [GetIniValues] which would resemble your program section to open the initialization file and place the data into the appropriate variable names.

When the [GetIniValues] section tries to input more values than are in the original file, an error occurs forcing the program flow to the [fixini] where the original values set by the user are put into the file along with the new default values. Then the program returns to [resumefromerror] where all the values can now be loaded into the variable names.

At the end of the demo, the \*.ini file will be deleted for you.

```
'Adding more values to an initialization file
'for newer program versions - Jim Grossman

'Create an original ini file for demonstration purposes.
'The values are put into an array only so you can see the
'progress of [GetIniValues]. Normally they would have value names

' In this demo I have called them value1$, value2$ etc.
open DefaultDir$+"\\MyIniFile.ini" for output as #MyIniFile
print #MyIniFile, "red":array$(1)="red":value1$="red"
print #MyIniFile, "blue":array$(2)="blue":value2$="blue"
print #MyIniFile, "green":array$(3)="green":value3$="green"
close #MyIniFile

Dim info$(10,10)
nomainwin
```

```
WindowWidth = 200
WindowHeight = 340
UpperLeftX=int((DisplayWidth-WindowWidth)/2)
UpperLeftY=int((DisplayHeight-WindowHeight)/2)

statictext #main, "Initialization Values", 45, 32, 150, 20
ListboxColor$ = "white"
listbox #main.lb1, array$(, [listboxClick], 55, 52, 100, 160
statictext #main.st1, "", 50, 227, 108, 20
button #main.b1,"Run Program", [RunProgram], UL, 30, 257, 150, 25

open "" for dialog_modal as #main
print #main, "font arial 10 bold"
print #main, "trapclose [Quit]"
print #main.st1, "Original Values"
wait

[RunProgram]
gosub [GetIniValues]
#main.st1, "Plus New Values"
array$(1)=value1$
array$(2)=value2$
array$(3)=value3$
array$(4)=value4$
array$(5)=value5$
#main.lb1,"reload" 'So you see all the values now.
wait

[Quit]
kill DefaultDir$+"\MyIniFile.ini"
close #main
end

'This is the part that goes into your program.
'It opens your ini file and gets the values or repairs
'the existing ini file.
[GetIniValues]
on error goto [fixini] 'When you request more values from the ini
file than exists, it will
'create an error and this line will redirect
program control.
'This can go early in your program.

'First look to see if an initialization file exists
'in case it somehow got deleted. If it did get deleted,
'rebuild it with all the default values including the values from
```

the

```
'current program version.  
'It won't be used in this demo since an ini file was created  
'at the beginning of the demo. Just good policy to check in my opinion.  
files DefaultDir$,"MyIniFile.ini",info$()  
if val(info$(0,0))=0 then 'If the ini files got deleted somehow, then rebuild.  
                                'with the original default values plus any new ones.  
        open DefaultDir$+"\\MyIniFile.ini" for output as #MyIniFile  
        print #MyIniFile,"red"  
        print #MyIniFile,"blue"  
        print #MyIniFile,"green"  
        print #MyIniFile,"normal"  
        print #MyIniFile,"yes"  
        close #MtabIni  
end if  
  
'If the ini file does exists, then open it and get values, but if an error  
'occurs because there aren't enough values,(wrong version), the error  
'will send the program to [fixini] because of the on error statement  
'at the beginning of the program. After [fixini], program control will  
'return to [resumefromerror] and all correct values will be retrieved.  
[resumefromerror]  
open DefaultDir$+"\\MyIniFile.ini" for input as #MyIniFile  
input #MyIniFile, value1$  
input #MyIniFile, value2$  
input #MyIniFile, value3$  
  
'The error will occur on the next line because there are only three values  
'in the original file. Program control will be directed to [fixini].  
'After [fixini] has run, all the values will be retrieved.  
input #MyIniFile, value4$  
input #MyIniFile, value5$  
close #MyIniFile  
return 'Then return from the gosub.  
  
'If an error does occur because of the wrong version ini file, thi
```

```
s section
  'will rebuild the ini file with the correct number of values using
  'values from the original version with the added default values fr
om later versions.
  'This section won't be used again until you add more values.
  [fixini]
  close #MyIniFile 'You must close the ini file because it was not c
losed above
                'when the error occurred.  Besides, it was opened
for input
                'and must now be opened for output.
  open DefaultDir$+"\\MyIniFile.ini" for output as #MyIniFile
  'Original version values.
  print #MyIniFile, value1$
  print #MyIniFile, value2$
  print #MyIniFile, value3$
  'Add later version values.  You can keep adding to this list as yo
u add more values.
  print #MyIniFile, "normal"
  print #MyIniFile, "yes"
  close #MyIniFile
  goto [resumefromerror] 'Then return from the error.
```