

## ACCESING TEXT FILES IN VISUAL BASIC FOR EXCEL

### 1. INTRODUCTION

Excel provides the capability to read and write text files. This is very useful since the input and output of several typical programs are text files.

Normally the way to access to these files is carried out with the old statements of visual basic. This document deals with the new way to have access to these files which is clearly more simple and efficient:

### 2. PROCEDURE TO ACCESS TEXT FILES

The first commands to read or append data on a file must be:

```
Set fis = CreateObject("Scripting.FileSystemObject")  
Set fi = fis.OpenTextFile("c:\document.txt", 1, 0)
```

Where fis and fi are variant variables. The variable fi will identify the file hereafter. The second parameter of OpenTextFile (1 on the example above) determines if the file is opened from the beginning and in read-only mode (1) or from the end to append additional data (8). The third value should be 0 in general to open as an ASCII file.

If the file is to be created or rewritten the following statements have to be used instead:

```
Set fos = CreateObject("Scripting.FileSystemObject")  
Set fo = fos.CreateTextFile("c:\document.txt", True, False)
```

Where fos and fo are variant variables. The variable fo will identify the file hereafter. The second parameter of CreateTextFile (True on the example above) determines if the file can be re-written if it exists previously (True) or not (False). The third value should be False in general to create as an ASCII file.

To read a line from the file the statement is:

```
cadena = fi.readline
```

To write a line on the file the statement is:

```
fo.writeline (cadena)
```

## ACCESING TEXT FILES IN VISUAL BASIC FOR EXCEL

Where cadena is a string variable that contains the data read or the data to be written.

In order to read all the lines on the file use the Atendofstream function

Do While fi.atendofstream <> True

```
    cadena = fi.readline
```

```
    .....
```

```
    .....
```

Loop

Finally remember to close the files when the input/output operations are ended:

```
fi.close
```

```
fo.close
```

The advantage of using this method is the additional functionalities provided by visual basic such as:

WriteBlankLines

SkipLine

Skip

ReadAll

Read

Write

Take care when using text files with the file format of the files. They have to be stored in PC format and not in UNIX format. If they are stored as UNIX files the end-of-line (EOL) character will never be found till the end of the file and the cadena variable will contain the whole file instead of one line.