

TAP INSTALLER

(Currently at Version B1.4)

1 Introduction

This is a Windows utility primarily developed to automate the installation of complex TAPs for “newbies” who have a PC running Windows, but also suitable to automate updating of Topy files by TAP developers. Its aim is to provide a more user friendly and flexible approach than a Windows batch file, approaching the ease of use of the standard Windows program installer.

The operation & user interface is entirely controlled by a script file, so can be tailored to suit any TAP.

It works with a direct USB connection to the Topy or with FTP transfer via a suitable server running Aldarin’s “FTP - Server for Topfield” such as Asus/Slug etc.

2 Main Features

A script file controls all the on-screen messages (including TAP name), and can:

1. Copy files (giving a % complete indication updated as the transfer occurs for direct USB mode only)
2. Delete files
3. Make directories
4. Delete directories
5. Display a “readme” type text file on completion, with hyperlink and email links.

with the ability to make the above operations dependent on user inputs or the presence/absence of files/directories. Up to 10 levels of nesting is permitted in such conditional statements.

In general the commands are entirely symmetric between Topy & PC, so one can copy in either direction and within either, delete files at either end, etc. The colour scheme and text sizes are all defined by the script file. A facility to display an initial “splash” screen specific to the TAP is also provided.

In addition to its normal mode, TAP INSTALLER has two modes to assist developers:

- A Demo Mode in which responses from the Topy are simulated. This allows TAP authors to initially evaluate its use, and then check their scripts without needing to have a Topy connection or actually copy any files.
- A Test Mode in which all Topy top level directories are amended to add a leading “T”, and thus a script command referring to “/ProgramFiles/..” is interpreted as applying to “/TProgramFiles/..”. The complete script can then be used to install in this parallel directory structure, leaving the normal use of the developer’s Topy unaffected.

Each mode is selected via a single command in the script file.

3 User Instructions

3.1 Basic Use

3.1.1. General

These instructions are intended as an outline of what will need to be published with the TAP to be installed, and assume that the TAP, Installer and the associated files will be downloaded from the internet as a “zip” file. They will need to be tailored to suit the particular application, need amendment if a self-extracting zip file is used, and have been written assuming use of IE rather than other browsers. MyStuff has been used as the example. Those aspects particularly likely to require change are the screenshots and those marked in yellow **thus**.

3.1.2. Preparation & Starting

Ensure your Toppo is not in Standby, and check that you can communicate with it using Altair (for USB connections) or your FTP program (for ASUS etc users). Ensure you close (not just minimise) Altair afterwards.

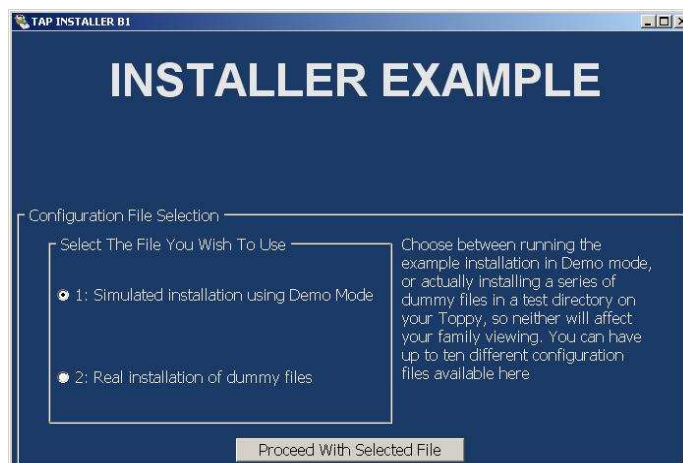
Download the TAP file from **{insert link here}**, and select the “open” file option. When the file has loaded, the Winzip (or equivalent) window should open. Scroll down the file to “Installer.exe”, and double click on this and Installer should start.

3.1.3. Initial Screens

Installer will start with a splash screen as follows:



Click on the message at the bottom of the screen (or press any key), **and you will then see a screen which allows you to select the operation you wish to perform as follows:**



(May want to expand on options here – also note this only applies if multiple configuration files are used see para 4.2 below)

Select the operation you wish to perform by clicking on the relevant radio button, and then click on the message at the bottom to select the next screen, which allows the user to input his configuration as follows:



Click on the “radio button” opposite your configuration, and then click on the label at the bottom to continue. If you select Windows XP and there is a connection failure, it will then try Windows Vista, and vice versa.

If you have selected the FTP connection, you will then see:



Enter the values for you connection, and click on the label at the bottom of the screen.

3.1.4. The Installation Process

Having completed the selection of the appropriate settings, Installer checks the connection to the Toppo, and then proceeds to copy the files to the right place. Messages displaying what has done will appear for approximately 2 seconds at each stage, and will report any errors encountered.

During the installation you will be prompted for Yes/No decisions concerning the following aspects of the installation:

- The source of Electronic Program Guide (EPG) data you intend to use.

- Optional skin and favourites files.

These are answered by clicking on either the “Yes” or the “No” appearing below the question. Where necessary the question is clarified by a message below the Yes/No buttons, and this clarification may include a hyperlink to more information on the topic. Simply click on the relevant link to open your browser at the relevant page.

On completion you will see a message indicating whether the installation has been completed successfully, and what you need to do next. This include hyperlinks to useful sources of further information etc and an email link for help requests . Simply click on the relevant link to open your browser/email program at the relevant page.

Pressing F1 toggles the display of detailed progress messages on & off – having them off speed up the process considerably.

3.1.5. Copying Between Topy Directories

If detailed progress messages are being displayed, you will see that when copying between Topy directories the files first have to be copied to the PC, and then copied back to the new location.

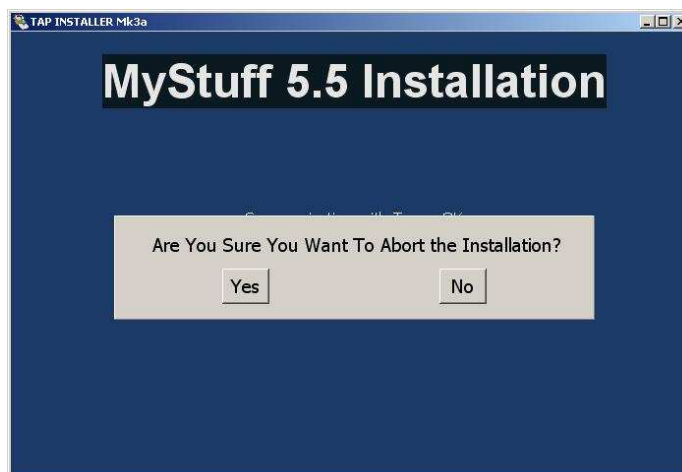
3.1.6. Log File

A log of operations generated as the operations are performed. This is a text file of which extracts can be sent to the TAP author for assistance with any problems arising. By default this is stored in the directory from which Installer was run, but this can be changed via the script file. The log file is called “xxxx_log.txt” where xxx is the TAP name as used in the title.

3.1.7. Standard Windows Controls

The user can stretch/compress the window and close the program at any time using the standard Windows controls. The latter will not undo any installation activities already started. Stretching the initial “splash” screen will not change the size of the image, however.

If the Windows “Close” button is clicked, further installation activities will be halted on completion of the current one, and an “Are you Sure ..” pop-up will appear thus:



Click on “Yes” to abort the installation, and on “No” to resume the installation

3.2 Advanced Use

3.2.1. General

For those regularly using Installer (e.g. TAP developers and testers) two additional features are available, as disused below.

3.2.2. Alternative Configuration Files

A valid configuration file with any name and extension can be used in place of the default "Installer.cfg" by either:

- Dragging & dropping the file onto Installer.exe in Windows Explorer.
- or:
- Using the command line `Installer.exe <path\anyname.anyextension>` either directly or within a batch file.

3.2.3. Ini File

A file called "TAP Installer.ini" is generated in Installer's directory whenever it is run, and if found when re-run will set the default settings to those used previously. This occurs when in use by the basic user, but he will be installing from a temporary directory and thus the ini file will not be available next time. However the advanced user is expected to keep Installer in a fixed directory and use it for tasks such as frequent updates of a few files, in which case the ini will be of use.

If the line "NoConfigScreens=0" in the ini file is manually changed to "NoConfigScreens=1" (e.g. using Wordpad), the initial configuration screens are suppressed and the ini values used. To get the configurations screens back, that line can be changed back, the line deleted, or the whole ini file deleted.

4 Information For Developers/Distributors

4.1 Directory Layout

When run, Installer.exe must be in the same directory as:

- The script file, called "Installer.cfg" unless using the facilities described in para 3.2.2 above.
- A copy of TFDLL.dll suitable for Windows XP users, renamed XP.dll.
- A copy of TFDLL.dll suitable for Windows Vista users, renamed Vista.dll.

with the following optional files:

- "Splash.jpg" if you want to have a splash screen.
- "Installer Success.txt" – a plain text file displayed on successful completion telling the user what to do next – reboot his Topy, read the manual, make a donation for example.
- "Installer Error.txt" – a plain text file displayed in the event of failure telling the user what to do next – post the log file on Topy.org.uk for example.

- Any additional script files.

Where possible, drive letters and directory paths should be specified in full, using drive “letter” “#” to denote the Topy. In the absence of a path (i.e. presence of ‘/’ or ‘\’) the directory from which TAP Installer is running is used (or subdirectories “Source” & Destination” if these exist) is assumed. If no drive is specified, the drive from which TAP Installer is running is assumed for the source of all copies, and the Topy as the destination for all other actions.

Note that copying from PC to PC may provide a very basic installation tool for PC applications if neither registry entries nor desktop short cuts are required.

4.2 Multiple Configuration Files

TAP Installer supports the use of multiple configuration files, so that the user can choose one of up to ten types of installation (or un-installation). In this case the “Installer.cfg” must include the line “AllowExtraCfg” and the TAP name, and can include “Skin data” and any common elements to all operations (e.g. making directory structure).

The additional files must be called Installer*.cfg, where the star represents any non-zero character string acceptable to Windows. In the options list they will appear as a filename unless they include the line “##Summary: ...’ in which case the (first 48 characters of the) subsequent text is displayed. The files are displayed in alphabetical order (sorted on first character only, numbers before letters) so the presentation can be tailored to give something like:



The spacing between entries is dependent on the number of files found, so is much less cluttered if the number of files is reduced.

The “ExtraCfgHelp: <Text>” command allows up to 240 characters of extra information about the choice of options to be displayed as shown in the example in 3.1.3.

4.3 Configuration File - Overall Customisation

The configuration file is a simple text file as produced by WordPad etc. Anything on a line after “##” will be treated as a comment and ignored. Only one command per line is allowed.

The file should always begin with the following command:

```
Title: <Text> // Default is "Unknown TAP"
```

and may also include:

```
SuccessMsg: <Text> // Default message is "Completed satisfactorily". This is displayed on successful completion of the installation if no "Success.txt" file is found.
```

```
FailMsg: <Text> // Default is "Sorry - not completed satisfactorily". This is displayed when errors have been found if no "Error.txt" file is found.
```

```
MsgTime: <Integer> // Time for display of progress messages in seconds, default = 2, minimum = 0
```

```
UseDemoMode // Selects demo mode
```

```
UseTestMode // Selects Test Mode
```

```
AllowExtraCfg // Allows multiple cfg files
```

depending on your needs.

4.4 Colours & Font Sizes

The appearance can be changed using the following “skin-type” commands:

```
Background=RGB(r,g,b) // Default grey
```

```
TitleText=RGB(r,g,b) // Default black
```

```
TitleBgd=RGB(r,g,b) // Default = Background
```

```
TitleFontSize: 35 // default is 26
```

```
ProgressText=RGB(r,g,b) // Default = TitleText
```

```
ProgressBgd=RGB(r,g,b) // Default = Background
```

```
ProgressFontSize: <integer> // Default is 12
```

```
HelpText=RGB(r,g,b) // Default = ProgressText
```

```
HelpBgd=RGB(r,g,b) // Default = ProgressBgd
```

```
HelpFontSize: <integer> // Default is 10
```

```
MessageText=RGB(r,g,b) // Defaults to Black
```

```
MessageBgd=RGB(r,g,b) // Defaults to light grey
```

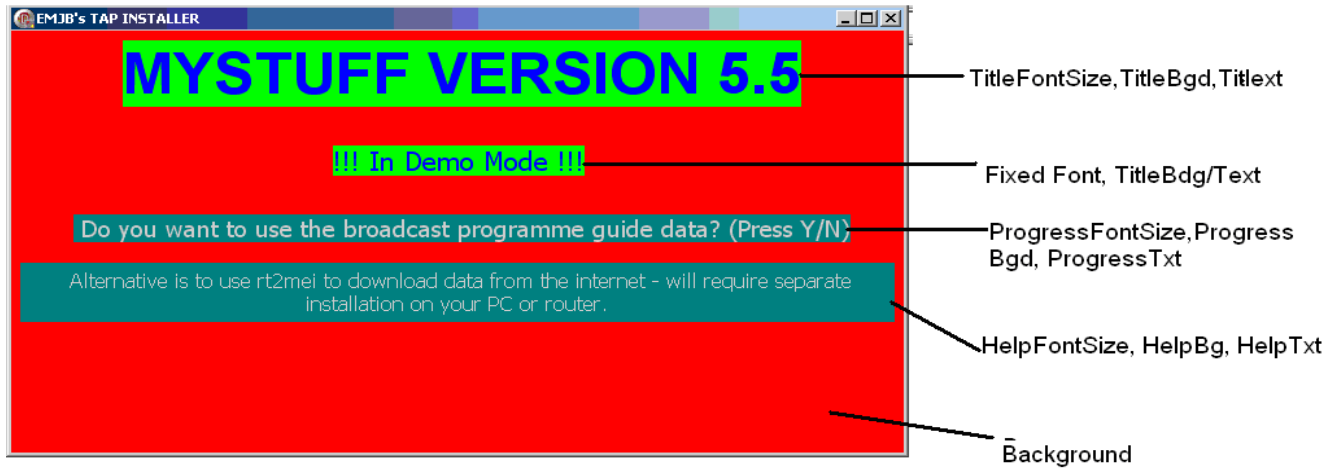
```
MessageFontSize: <Integer> // Default is 12
```

```
Width:<Integer> // Defines width of display in pixels – default 696
```

Note that when a value is set all others using it as a default are set to that value even if they have previously been set to another value. Thus Having “Background=RGB(x,y,z)” at the end of the list would overwrite previous values of TitleBgd, ProgressBgd, and HelpBgd so it is recommended that you keep to the above order.

The Help values are also used for the final text screen, if used.

The various fields above are explained by the following (revolting) colour scheme:



The “Message” field is that used to display the final success/failure messages. Most other screenshots are with the default or MyStuff settings.

Unfortunately there seems to be a problem “skinning” the various control buttons etc, so these are always black on light grey.

4.5 Script File – Functional Commands

4.5.1. General

Files/directories on the Topyy are denoted by ‘#’ as a pseudo drive letter – e.g. “#:\ProgramFiles\”. In the absence of a path, the directory from which installer was run is assumed.

Forward and backward slashes are generally interchangeable in file paths, but not in the “//” in hyperlinks. In common with Altair, the PC backward slash notation is used in the displayed reports whether referring to Topyy or PC directories.

File & directory names on the PC are not case sensitive, but those referring to the Topyy are. However when copying from PC to Topyy, the case used in the PC source filename will be carried across to the Topyy.

Leading spaces or tabs can be used to improve the legibility of the file.

4.5.2. CopyFile:

This command takes the form of:

CopyFile: <source> **to** <destination>

where <source> & <destination> include the full path unless the file is (to be) in the directory holding Installer.exe. Wildcards (* & ?) can be used in the source filename, but not in the path. The destination is considered to be a directory with the filename to be copied taken from the source if the destination ends in “\” or “/”.

N.B. There is currently only limited support for wildcards when copying from Topy to PC or Topy to Topy using FTP – “?” is not supported and “*” can only be used for the complete filename or extension so “X.*”, “*.X” will work but “X.t*” will not .

A typical TAP installation command might be:

CopyFile: MYTAP.tap to #:\ProgramFiles\Auto Start\

The action is considered a failure if the source file cannot be found, or is not successfully transferred. If the file already exists in the destination directory it will be normally be overwritten. However

:/NoOverwrite

can be added at the end of the command if an existing file of that name is not to be overwritten

4.5.3. DeleteFile:

This command takes the form:

DeleteFile: <path & filename>

e.g.

DeleteFile: #:/ProgramFiles/Settings/MYTAP.ini.

The action is considered a failure if the file cannot be found, or if the delete action fails. Wildcards (* & ?) can be used.

N.B. There is currently only limited support for wildcards when deleting Topy files using FTP – “?” is not supported and “*” can only be used for the complete filename or extension so “X.*”, “*.X” will work but “X.t*” will not .

4.5.4. Mkdir, Rmdir

These take the obvious forms:

Mkdir: <path>

Rmdir: <path>

Note that only one level of directory can be created at a time, so for example “#:\ProgramFiles\Settings\Skins” can only be created if “#:\ProgramFiles\Settings” already exists – other wise a failure will be reported. Only empty directories can be deleted, and wild cards are not permitted.

4.5.5. UserConditional:

This command allows branching according to whether the user presses “Y” or “N” in response to a scripted question. The syntax is:

UserConditional: <Question text>

Help: <Text> (Optional)

.

.

Else:

.

.

EndCondition:

The “Else” statement can be omitted if nothing follows. An example might be

UserConditional: Do you want to load some alternative “skin” files (Press Y/N)?

Help: These provide alternative colour schemes for MyStuff and some other TAPs

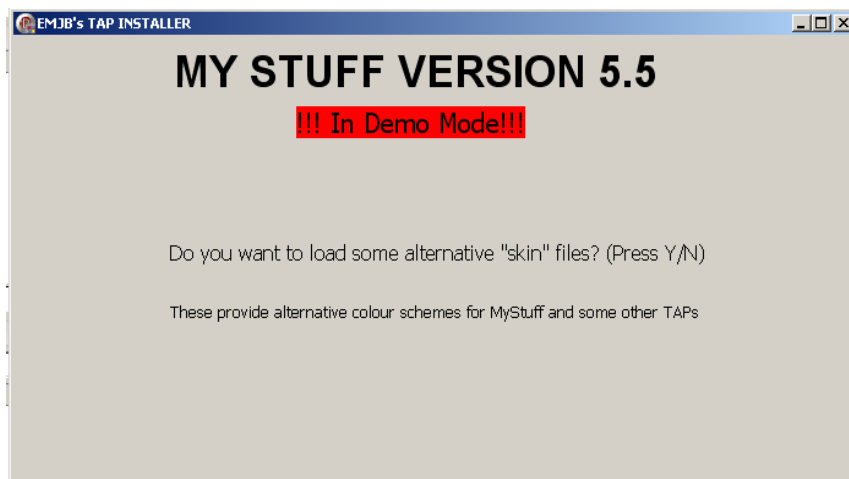
MkDir: #:\ProgramFiles\Settings\Skins

AddFile: *.mcf to #:\ProgramFiles\Settings\Skins

EndCondition:

The “**HELP:**” line can be omitted if not required. It appears underneath the question to help the user understand the implications of the question. It has a maximum length of 300 characters. The command can be inserted anywhere up to the “EndCondition:” command.

The above might appear as:



A total of up to 10 levels of nesting can be used with the “UserConditional:” & “FileConditional:” commands. N.B. A conditional of one type within the other type counts as two levels.

4.5.6. FileConditional:

This command allows decisions to be made on the basis of whether a file is present, and takes the form of:

FileConditional: <File path & name>

.

.

Else:

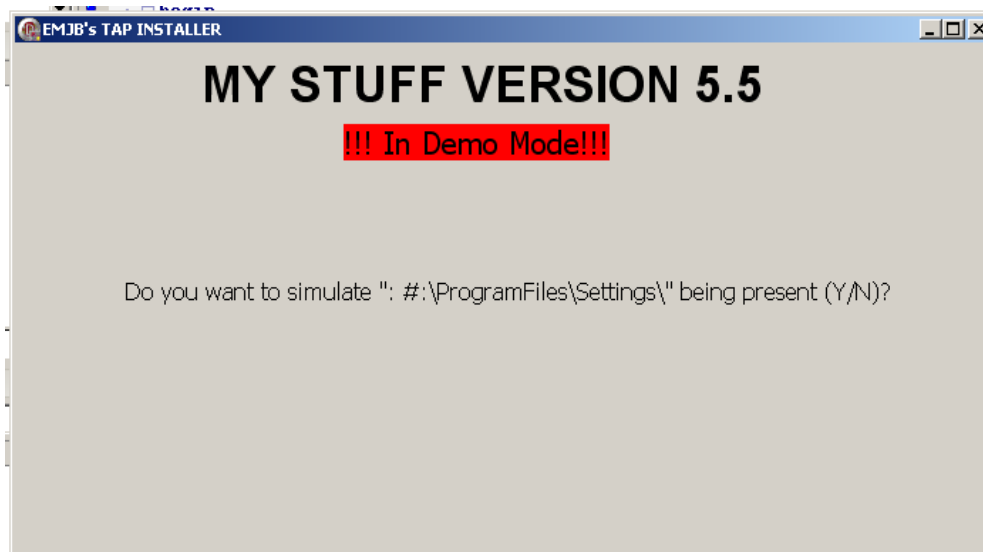
.

.

EndCondition:

An explanation of the operation can be displayed by using the “Help:” command.

Clearly in Demo mode, this command cannot interrogate the Toppo to see if a file is present, so “**FileConditional:** <File/Directory path/name>” is automatically replaced by “UserConditional: Do you want to simulate the presence of < File/Directory path/name> on the Toppo(Y/N)?” thus:



Wildcards (* & ?) can be used in filenames and in the lowest level directory.

N.B. There is currently only limited support for wildcards when interrogating the Toppo files using FTP – “?” is not supported and “*” can only be used for the complete file name or extension so “X.*”, “*.X” will work but “X.t*” and “X*.tap” will not .

4.5.7. DirConditional:

This is the directory equivalent of “FileConditional”. Note that wildcards can only be used in the lowest level directory part of the name, so “/*.* /Settings” is not legal but “/Settings/*.*” is legal.

4.5.8. “ShowMessage” Command

This is intended to provide summary indications of what is happening when MsgTime is set to zero. It generates a message that appears in the “Progress” and/or “Help” areas of the screen, e.g. to provide a commentary on what is going on. The syntax is:

ShowMessage: <Progress text> && <Help text> [Display time]

If nothing is to be displayed in the help field, the “&&” can be omitted. The Display time is a decimal number, and is the minimum time in seconds for which the message is displayed. Other actions are suspended until the time expires. The message will continue to appear until the next message is generated.

4.5.9. “LoadFw” Command

This allows new firmware to be loaded into the Toppo. When used with FTP, the whole process is automatic including a reboot, but with a direct USB connection the user has to manually reboot the Toppo – the reboot command below will not work once the LoadFw command has been issued. The syntax is:

LoadFw:<File path & name>

The file name must have a “tfd” extension, and this command should always be at the end of the configuration file.

4.5.10. “ResetTopy:” Command

This reboots the Topy, and if used should be the last command in the script file.

4.5.11. Changing the Names of “Installer Success.txt” and “Installer Error.txt”

The names used when looking for the final files can be changed using the commands:

SuccessTxtFile: <File name>

ErrorTxtFile: <File name>

This enables use of a file called “readme.txt”, for example, or for the displayed files to be changed when using multiple configuration files. These commands could also be included within “Conditional” statements to reflect the options the user has selected.

4.5.12. Logging

The command

SaveLog to: D:\Temp

will save the log file in the indicated directory, rather than in the directory from which Installer was launched. The default directory has the advantage of not “polluting” the user’s PC, but can easily lead to it being accidentally destroyed. No file will be saved if the indicated directory does not exist, so it should be preceded by the relevant Mkdir command(s).

4.6 Hyperlinks

Hyperlinks can be included in the UserConditional “question”, in the “help” instructions, and in the final text screen. Email links can be included in the final text screen only.

Hyperlinks must take the form of “http.....” followed by a space. Email links must take the form of <mailto:.....> followed by a space with the option of space between the “[mailto:](mailto:.....)” and the address.

Use of these links does constrain the colour scheme, however. Unless the background colour is royal blue, the links will always appear royal blue and so will look very odd with some schemes. With a royal blue background, the links are white.

4.7 Renaming Installer

You may rename Installer to something more suitable to your use thereof if you so wish, but must retain the “.exe” file type.

5 Run Sequence

When the installer runs, the initial operations are:

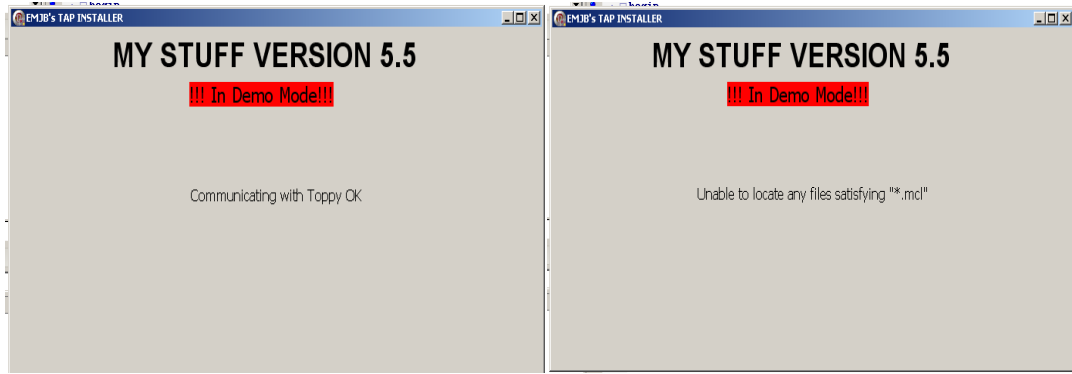
1. If a splash screen file (Splash.jpg) is found, this is displayed until any key is pressed or the user clicks on the associated message, thus:



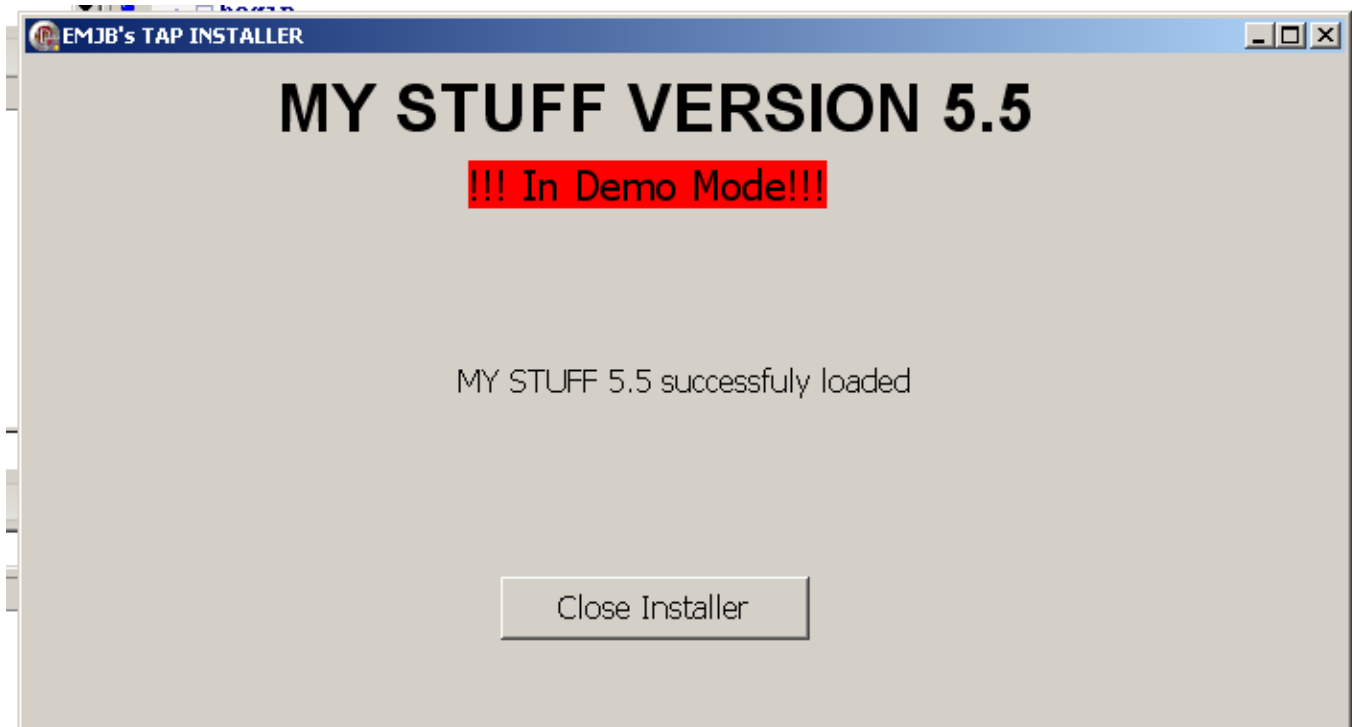
2. Initial syntax check on the basic script file, displaying the whole line in the event of an error being found. If satisfactory the skin colours etc are taken for use until any change in a supplementary file.
3. If the initial cfg file allows further files, Installer looks for these and displays a list if more than one is found for the user to choose.
4. The chosen supplementary cfg file is checked for syntax
5. Check that there is a TFDLL.dll file in the current directory for USB transfers, or that a successful log-on occurs in FTP mode.
6. Check that it can communicate with the Topyy, by looking to see if the “\ProgramFiles\” directory exists.

If the last check fails in USB mode, the user is encouraged to check whether the Topyy is switched on, the USB cable connected, and Altair closed down. When the problem is fixed, the installation will continue. In FTP mode the user is merely advised to check the Topyy is on and connected, but cannot continue simply by rectifying the situation.

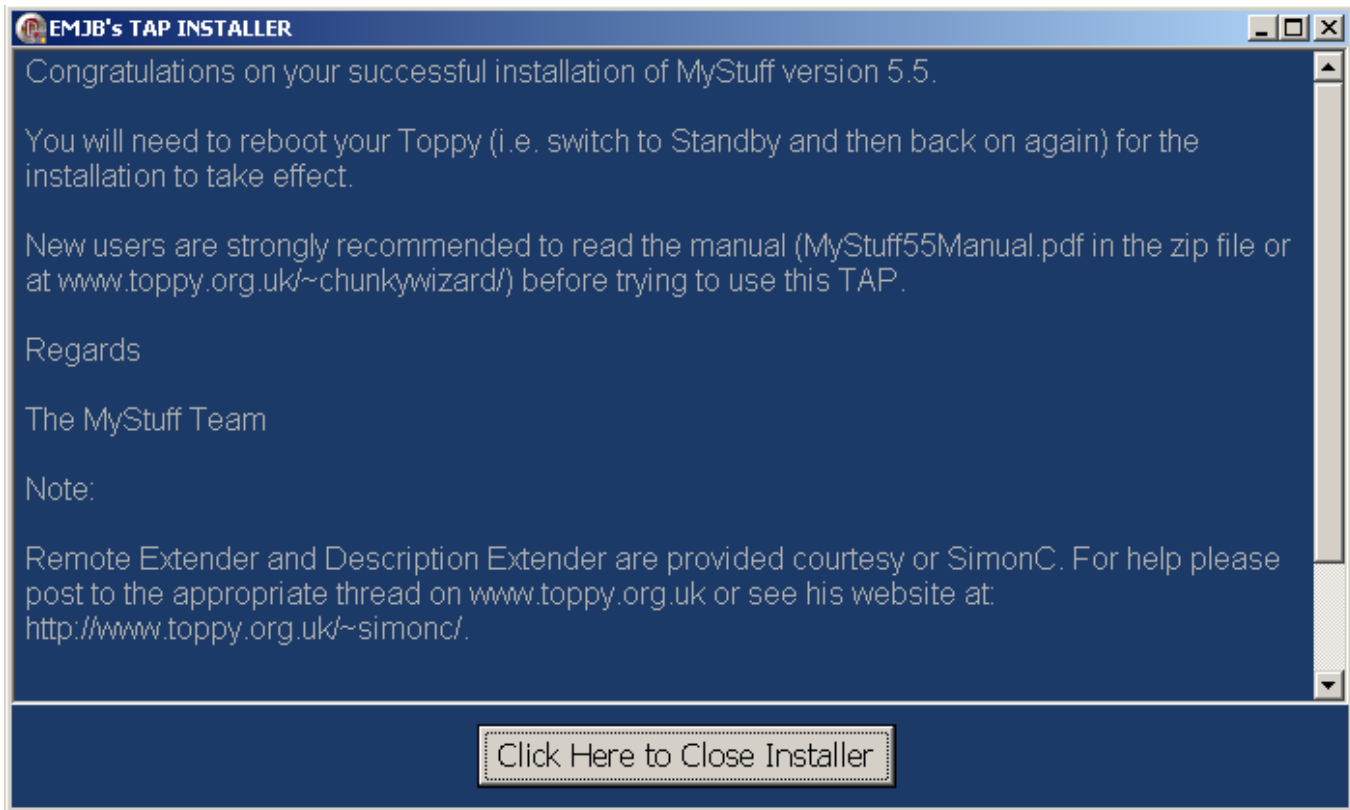
Thereafter as each action completed a message containing a description of that action is displayed for a script-defined time. If an action fails, the subsequent actions are abandoned, a failure message displayed, and the user is invited to close the program. If all actions are completed satisfactorily, a success message is displayed, and again the user is invited to close the program. The criteria for success of each action are clarified in section 4.5 above where there is any doubt. Examples of OK & failure progress messages are given below:



and the final stage:



or one can display a text message such as:



The scroll bar on the right enables the user to scroll down where the message occupies more than a page, or the user can click on the relevant window and scroll up/down with a wheel-mouse.

6 History

B1: Released for use with MyStuff 5.5 installation

B1.1: Enhancements for use of multiple configuration files.
Allows use of "/Source" & "/Destination" subdirectories in the installation directory

7 Acknowledgements

To Aldarin, for use of his Delphi library, without which I would not have even dreamt of starting on this project, Bawbagg for his help in general and specifically for the FTP drivers, AKPetrov for the hyperlink logic, and to ChunkyWizard for his encouragement and example files used in the above screenshots.

Last Updated: 2/5/2008