

Windows Application Installer

Some brief notes about this application. This was originally written to help a distinguished contributor to BIX in installing applications in the Microsoft Windows Program Manager when building systems for his clients. The installation mechanisms supplied with Windows are too cumbersome to use when installing tens of applications on system after system, day after day, especially when no two systems are alike.

This application is capable of installing a list of icons in Program Manager groups in a matter of seconds. It can run in a fire-and-forget mode: with the proper options enabled, when it is started it will execute the installation and terminate when complete, without any further user input. It takes instructions from a script file written using the Windows "Profile" file conventions. Examples of such files are WIN.INI and PROGMAN.INI in the Windows directory. The instructions in the file specify the options enabled, the Program Manager groups to be created, the applications to be installed in those groups, and any applications to be named in WIN.INI for automatic startup when Windows first runs.

Profile syntax is as follows:

```
[Topic]
key=value
key2=value
etc.
[AnotherTopic]
key=value
etc.
```

Applications can determine the value of a key under a topic using a simple function call. The name of a key may not contain embedded spaces, but the value, if a string, may be any combination of characters and spaces. Integer values are also allowed. A keyname must be unique within a topic, but may be duplicated in other topics. Topic names must be unique.

The script file for this application **MUST** be called **APPMAN.INI**. It must reside in the same directory as this application. Here are the topics which are found in the file.

[Appman] - This topic's keys set or clear various options. The topic is optional: if absent, all the options default to "off" or zero. These values can be toggled using the menu in this application.

[Groups]-The keys in this topic are defined by the user and refer to other topics defined later in the file. For instance if a key/value pair is "MyGroup=1" then it refers to topics [MyGroup] defining the Program Manager group to be created, and [MyGroup-apps] listing keys of applications to be installed in the group.

[WINI.INI-load] - This topic is a list of application keys that refer to application topics elsewhere in the file. These applications will be added to the "load=" line in WIN.INI. This topic is optional.

[WINI.INI-run] - This topic is a list of application keys that refer to application topics elsewhere in the file. These applications will be added to the "run=" line in WIN.INI. This topic is optional.

Minimum contents for the file: the [Groups] topic with at least one group key, then the topic for that group's applications (see later) containing at least one application key, then the topic for that application, containing at least the File key.

Keys in [Appman] - these values may be toggled using the Installer's Options menu.

AutomaticInstall - If value = 1, the installation runs as soon as the application is launched, unless it is launched as an Icon (hold down SHIFT key when launching). If absent, or value = 0, the installation will not proceed until you start it from the application menu.

AutomaticStop - If value = 1, the application will terminate as soon as the installation completes. If zero or absent, the application will display a completion message for acknowledgment, then remain active with further installations disabled. This allows inquisitive

users to examine the application, view the About box, and in future versions, change settings via menu or dialog boxes.

AutomaticLoadRunSetup - If value = 1, any applications specified for the WIN.INI load and run lines will be added without user intervention. If absent or zero, the user has to approve the new versions of those lines via dialog boxes.

Group key names

These are user defined. Invent a name (no embedded spaces) for each group to be created in the Program Manager. Put that as a key name under the **[Groups]** topic and give it a value, preferably integer, the actual value being unimportant. Later in the file, define two topics: **[keyname]** and **[keyname-apps]** where "keyname" is the name you invented. Under **[keyname]** you can put the following key/value pairs:

Caption=string - String is any set of characters, including spaces. Quotes are not necessary. This will be used as the title of the Group Window in the Program Manager. If not supplied, or this whole topic is absent, it defaults to the "keyname".

File=filename - A valid DOS filename. No quotes needed. This is optional. If supplied it will be used as the name of the group file created by the Program Manager. If absent, Program Manager will invent a name. This is not often used.

Under **[keyname-apps]** you put a list of application keynames which you invent. Give each one a value, preferably integer, value unimportant. Each application keyname corresponds to an application topic defined later. Same rules as before: no spaces in keynames.

Application key names.

Invent a key name, e.g. appkeyname, for an application topic, and put it in one or more of the group keyname topics, or the WIN.INI topics. Then define the application topic, e.g. [appkeyname] and put in the following keys:

File=DOS pathname. This is required for an application. The pathname may optionally include disk drive, directory path. Filename and extension are required. The string is passed unchanged to Program Manager. If drive and path are not supplied, Program Manager will look for the file using the PATH environment variable.

Caption=string. Same as for a group caption. Any characters allowed, including spaces. This will be used as the Title of the application Icon in the Program Manager Group. This is optional: if absent, the application filename will be used.

IconFile=DOS pathname. If you want the Icon representing the application to come from another file, or it is not a Windows application, put a name for the file here. The file may be an executable, icon file (.ICO), a DLL, or any other file containing Windows icon resources. This entry is optional: if absent Program Manager will get the icon from the application or use one of its default icons, whichever is appropriate.

IconIndex=integer. Sometimes an application or the **IconFile** has more than one icon in it. The default one to show is numbered 0. Extra icons will be numbered 1,2,3 etc. If this entry is absent it defaults to 0. If you specify an index that Program Manager cannot find an icon for, it will use one of its default icons.

Errors in scripts:

If the [Groups] topic is empty, or absent, then the Installer will report an error and terminate.

If the application key list for a group is empty or absent, the group will not be created.

If the File value for an application is absent or invalid, the application will not be installed.

If the Program Manager reports an error on an operation (create a group, add an application) that operation is skipped and the next one is performed. This may lead to strange results if, say, a group creation fails: any applications installed after that will go into the previously active group. This will be tightened up in a later version.

You can put in as many application topics as you like. One application file may be in several application topics, defining its title, icon etc. in different ways. One application topic may

be referenced in one or more group application lists, as well as the WIN.INI application lists. An application topic may be created for future reference and not used anywhere else in the file, if desired.

For an example of the syntax, see the APPMAN.INI file supplied with the software. This file is functional. If you wish to test the Installer using this file, you can easily remove the groups created in Program Manager using the Delete function in Program Manager.

Comments, suggestions etc. should be addressed to "davejones" on BIX, the Byte Information Exchange.

Dave Jones
a.k.a. Ayresome Associates.