

K-INSTALL for DOS and Windows 2.0

Copyright (c) 1994-95 by Rob McDonell, ARK ANGLES
All Rights Reserved

DEVELOPER'S MANUAL

INTRODUCTION

K-INSTALL is one of the most powerful and flexible installation programs around, in a very robust and compact package. It comes in both DOS and Windows versions. Both versions feature a wide set of almost identical capabilities such as:

- * checking CPU type and maths coprocessor,
- * checking DOS and Windows version,
- * deleting files prior to installation,
- * installing from and to any drive and directory
- * installing within the same directory
- * installing files to several different drives and directories,
- * optionally replacing or appending to existing files,
- * prompting for additional disks as necessary,
- * asking whether to install optional components,
- * unpacking LZH, ZIP, ARJ and self extracting archives,
- * expanding compressed files,
- * creating or updating AUTOEXEC.BAT and CONFIG.SYS,
- * creating or modifying WIN.INI, SYSTEM.INI and other INI files,
- * creating or modifying a file with registration data,
- * updating Windows Program Manager,
- * running external programs before, during and/or after installation,
- * linking to an external product catalogue,
- * linking to an external help facility,
- * providing messages and a status bar as installation proceeds,
- * rebooting DOS or restarting Windows after installation,
- * giving additional instructions at the conclusion of the program,
- * running the main program from the installation program,

- * uninstalling the program again,
- * modifying display colours, layout and graphic,
- * auto-starting any or all of the above functions.

and much more. And it's all controlled from a single settings file.

This manual describes both the DOS and Windows version of K-INSTALL Standard Edition. The few differences between them are indicated where appropriate.

System Requirements

K-INSTALL for DOS will run on any IBM-PC or compatible computer with 256KB memory. DOS version 3.0 or higher is recommended.

K-INSTALL for Windows will run on any IBM-PC or compatible computer running Windows 3.1 (preferably in Enhanced Mode if unpacking archives).

Installation Instructions

K-INSTALL installs itself! Run INSTALL.EXE or SETUP.EXE to unpack and copy the whole package. You get the installation program itself, documentation, and samples to show the full power of this versatile piece of software.

Support and License Information

Individual registration entitles the registered user to free support for at least 12 months, and a license to use and distribute INSTALL.EXE and/or SETUP.EXE royalty free with any software developed by that individual.

Business registration entitles the organisation to free support for at least 12 months, and a license to use and distribute INSTALL.EXE and/or SETUP.EXE royalty free with any software developed for it by any number of its employees from the one site.

Customised versions of K-INSTALL are also available such that options can be hard coded into the program rather than specified in the separate settings file. This not only makes the program even more secure, but also smaller.

Registered users get prompt resolution of problems, and free upgrades if they encounter bugs.

Warranty

ARK ANGLES warrants the physical disks and related printed matter to be free of significant defects in materials and workmanship, and that the program works largely in accordance with its documentation. This warranty is effective for three months from date of registration, and is not transferable.

No liability is accepted for any loss, damage or inconvenience caused by the use of this product. No warranty is made that it is suitable for any particular person or purpose.

In any event, the liability of ARK ANGLES is restricted to either a refund of the registration paid, a replacement of the physical disks and/or printed matter, or the provision of an upgrade to the latest version.

Further Information

For more information see the following files on the Registered Installation Disk:

README.TXT Support and license information, program history.

REGISTER.TXT Registration and pricing information

PROGRAM DESCRIPTION

K-INSTALL is a flexible and powerful program. There are many options to tailor it to run just as you need it. However you can also get the basics up and running very quickly. The INSTALL or SETUP executable file can be renamed if required.

The Settings File

K-INSTALL uses a settings file to control its many functions. File names, directories, and many other options are defined by special keywords. Suitable default values are used for any options not explicitly defined, ensuring that the settings file can be extremely simple.

Creation and Use of Directories

K-INSTALL can be run from any drive or directory. When running on DOS version 3.0 or later it will automatically identify its own location and look for all other installation files in the same place. On DOS versions earlier than 3.0 it will assume it is running from A:\. For multi disk installs the user can change the "from" drive and/or directory for each new disk, allowing faster installation using two drives alternately, or installation from multiple directories on a hard disk.

K-INSTALL automatically asks for and creates a main directory in which to install the program. If this directory already exists (unless it is a root directory) confirmation is obtained from the user before continuing the installation. Extra directories can also be created and/or accessed, including subdirectories of the main directory, the boot directory, the DOS directory, the Windows directory, the Windows System directory, and the drive and directory from which the program is being installed.

K-INSTALL can also be set to append to files, always overwrite/replace existing files, only overwrite/replace older files, or never overwrite existing files.

K-INSTALL can install a program within the same directory, automatically identifying files that do not have to be copied, while still unpacking archives and expanding compressed files. This is particularly convenient for users who obtain software electronically, as they do not have to use diskettes or other subdirectories temporarily to be able to install the program.

The default directory where the program is to be installed is specified with the DIR keyword in the settings file. The other directory options are controlled through INSTALL and MKDIR keywords.

Archives and Compressed Files

K-INSTALL automatically unpacks any files with extensions of .LZH, .ZIP, .ARJ, .EXE and one definable extension if the files have the unpack identifier in the file name. The unarchiving programs used are LHA.EXE, PKUNZIP.EXE and ARJ.EXE respectively. The options parameters passed to these unarchiving programs when they are run may be altered by the LHA, ZIP and ARJ keywords. Parameters for self-unpacking archives are set with the EXE keyword.

The unpack identifier is normally a single underline, which means that files such as _ABC.ZIP or MY_FILE.LZH ARCHIVE_EXE will be unpacked automatically. The unpack identifier can be changed in the settings file with the UNPAK keyword. If you do not wish any files to be unpacked then it is a good idea to change the identifier to characters that are invalid in file names, such as "<" or ">" or ":" etc.

K-INSTALL also automatically expands COMPRESSED files (created by the program COMPRESS.EXE) where the last character of the file extension is a single underline. This is not changed by the UNPAK keyword. If the file was compressed with the -r or /r parameter then the original file name is restored when the file is copied, unless a specific new name is given by the INSTALL keyword.

If you wish to unpack other types of archives (such as .ARC, .PAK or .ZOO files), or you want to use a different unarchiving program for .LZH, .ZIP and .ARJ files (such as UNZIP.EXE instead of PKUNZIP.EXE), then a file

extension and associated unarchiving program can be defined with the ARCHIVE keyword.

Appropriate versions of the unarchiving program(s) must be accessible to K-INSTALL to enable it to automatically unpack any archives that are not self-extracting. Also, in the DOS version of K-INSTALL (but not the Windows version), the program EXPAND.EXE must be accessible to enable automatic expansion of compressed files. This means the decompression program(s) must be either somewhere in the user's DOS Path (but this cannot usually be relied upon), or on the installation disk itself. For multi disk installs the decompression program(s) should actually be installed from the first disk to the main program directory, from where they will be available to unpack archives on subsequent disks, otherwise the decompression program(s) would need to exist on every disk that contained an archive or compressed file.

Note that multi-volume archives are not supported.

User Data

At the commencement of the installation process K-INSTALL asks the user for the name of the directory in which to install the program. The default value for this is specified through the DIR keyword, which can also specify whether the user has the option to change the directory, and how the question is asked.

The user can also specify whether to install certain optional components of the package. This is controlled by INSTALL keywords.

Most importantly, K-INSTALL can be set up to request other information from the user such as serial number, name, company, address, even configuration preferences. The information can be validated for completeness and correctness, which can be particularly useful to ensure the user has a genuinely registered package and knows the required format of data such as the program serial number. Entry of user information is specified through INPUT keywords.

The actual data entered by the user, along with certain other variable data such as file and directory names, can be used in the AUTOEXEC.BAT file through the PATH, AUTO and SHELL keywords, in the CONFIG.SYS file through the CONF keyword, in the registration data file through the RFILE and REG keywords, in INI files through the IFILE and INI keywords, in the Windows Program Manager information through the GROUP and ITEM keywords, in all external programs through the CATALOG, HELP, FIRST, LAST, INSTALL, LHA, ZIP, ARJ, EXE and ARCHIVE keywords.

Updating System Configuration Files

Many options are available to modify the configuration of the user's computer. This includes updating the DOS Path in AUTOEXEC.BAT, and the FILES and BUFFERS value in CONFIG.SYS. Any number of other lines may also be added to both of these files. These options are enabled through the PATH, AUTO, SHELL, CONF, FILES and BUFFERS keywords.

Additional options are available to create or modify the Windows WIN.INI, SYSTEM.INI or private INI files. This is done through the IFILE, ISECT and INI keywords. Program Manager can also be updated with the GROUP and ITEM keywords.

All these functions can include user entered data as defined in INPUT keywords.

For safety, backup copies of AUTOEXEC.BAT, CONFIG.SYS and INI files are taken prior to modification, and the original versions are automatically restored if the modification fails for any reason.

Creating Registration Data Files

K-INSTALL can create or modify files with any combination of predefined and user entered data. This is much more flexible than standard INI file creation, and allows complete control over the format of the data in each file. It is possible to create a registration data file which validates that the program is a genuine registered version, a configuration file which defines the various characteristics of how the program will run, even a batch file to run a program with user-defined parameters.

The files may be in plain Text format, or machine readable data files using either C format (null terminated strings) or Pascal format (length byte followed by string) or Binary format (with neither a length byte or a null terminator). The Text format allows lines to be written or appended to INI-type files, batch files or other configuration files. The C or Pascal or Binary formats allow data to be written directly into EXE files or DLLs or the like.

The creation or modification of these files is done using the RFILE and REG keywords, often in conjunction with INPUT keywords. The data can be written to the installation disk and then copied as part of normal install procedure, or written directly to the target directory.

Uninstallation

K-INSTALL can keep a log of all new files written and all directories created, as well as most changes to AUTOEXEC.BAT, CONFIG.SYS, INI files and Windows Program Manager. The uninstall function can be run immediately or at any time in the future from the same install program. It will delete any files that are still in the same location, and then remove any directories that are now empty again.

Confirmation is requested from the user before deleting files from the Windows or Windows System directories, as these files may subsequently have been used by other programs as well.

Files that have been created by the application after installation, such as saved desktops or configuration files, can be deleted through the REMOVE keyword.

Uninstall also undoes most changes to AUTOEXEC.BAT, CONFIG.SYS, INI files and Program Manager.

Users appreciate an uninstall facility with their software, especially for Windows applications, so it should be highlighted as a major feature of your programs. It is a good idea to copy INSTALL.EXE or SETUP.EXE and at least a basic settings file to the main installation directory, and then add an Uninstall icon to Program Manager which runs K-INSTALL. SETUP.EXE includes an icon number 2 which can be used for un-installation.

Additional Information

The appearance of K-INSTALL can be highly customised in terms of screen layout, colour and text. This is done through the TITLE, MSG, COLS, WINDOW, PATTERN, PICTURE and ICON keywords in the settings file. The CATALOG keyword can also be used to link a separate catalogue or other information file into the main install screen.

While K-INSTALL is installing the program it provides information about each file copied or unpacked, and can also display a progress indicator showing how much has been completed. The progress indicator is controlled through the file sizes defined with INSTALL keywords in the settings file. The progress indicator is as accurate as the sizes entered on the INSTALL record(s).

When K-INSTALL terminates it provides final instructions on the status of the program, how to run it, and where to get more information. This is customised through the GROUP, RUN and DOC keywords in the settings file.

PROCESSING

The Main Screen

The main K-INSTALL screen displays a number of functions, depending on the keywords specified and the state of the program being installed.

- * The Help function is displayed if defined by a HELP keyword and if the help program can be found.
- * If there is a log file in the directory from which K-INSTALL is run, then the program has already been installed so the Uninstall function is displayed. Otherwise the Install function is displayed.
- * The Reboot function is displayed if the program has been installed and the level defined by the RESTART keyword is 2, or the level is 1 and AUTOEXEC.BAT or CONFIG.SYS have been modified.
- * In the Windows version only, if the program has been installed and there is no Reboot function then the Restart function is displayed if the level defined by the RESTART keyword is 2, or the level is 1 and WIN.INI or SYSTEM.INI have been modified.
- * If there is no Reboot or Restart function then the Run function is displayed if there is a program defined by a RUN keyword and that program can be found.
- * The Catalogue function is displayed if defined by a CATALOG keyword and if the catalogue program can be found.
- * The Exit function is always displayed.

The FUNCTION keyword can be used to automatically start any or all of the above functions if they are available.

Installation

When the user initiates the installation process within K-INSTALL, the program performs its steps in the following order:

1. Checks the CPU type if required by the CPU keyword.
2. Checks whether a coprocessor is installed if required by the COPR keyword.
3. Compares the DOS version if required by the DOSVER keyword.
4. Compares the Windows version if required by the WINVER keyword.
5. Asks the user to enter the directory into which to install the program. If the directory already exists, checks that the user wishes to continue the installation.
6. Checks the total program size if defined by INSTALL keywords against the target drive.
7. Asks the user about any optional components if required by INSTALL keywords.
8. Asks the user to enter any variable data if required by INPUT keywords.
9. Creates the main program directory.
10. Creates any directories specified by MKDIR keywords.
11. Runs a program if required by the FIRST keyword.
12. Deletes any files specified by DEL keywords.
13. Creates a registration data file if required by RFILE and REG keywords.

14. Processes the list of files defined on INSTALL keywords, prompting for new disks if necessary, copying or unpacking files as appropriate, and then running any specified program.
15. Updates AUTOEXEC.BAT if required by PATH, AUTO and SHELL keywords.
16. Updates CONFIG.SYS if required by FILES, BUFFERS and CONF keywords.
17. Updates INI files if required by IFILE, ISECT and INI keywords.
18. Updates Windows Program Manager if required by GROUP and ITEM keywords.
19. Runs a program if required by the LAST keyword.

Uninstallation

When the user initiates the uninstallation process, the program performs the following steps:

1. Deletes any files specified in REMOVE keywords.
2. Deletes each file listed in the log file (that is, those files that did not exist prior to installation).
3. Undoes most changes to AUTOEXEC.BAT, CONFIG.SYS and INI files.
4. Removes icons from Program Manager (Windows version only).
5. Removes each directory listed in the log file (that is, directories created by installation) if those directories are now empty.

THE SETTINGS FILE

The settings file is a plain ASCII file that can be created and updated with any editor in DOS or Windows.

The default settings file has the same name as the installation program itself with an extension of ".SET". Therefore if the installation program is SETITUP.EXE then the settings file must be called SETITUP.SET.

A different settings file can be used by entering its name (as a parameter when running K- INSTALL. For example,

```
INSTALL MYSET.TXT
```

will read the settings from "MYSET.TXT". If no file extension is entered then ".SET" is presumed. Do not enter a directory name, as the settings file must always be in the same directory as the installation program.

Each option is defined in the settings file by a single keyword, often followed by one or more additional parameters. For example, to specify that the main program directory is to be added to the DOS Path in AUTOEXEC.BAT the line:

```
PATH
```

is added to the settings file. To specify the default directory in which to install the program a line like:

```
DIR C:\GAMES\BLASTER
```

or similar is added. Or to describe files or groups of files to install, include lines such as:

```
INSTALL READ.ME, ., 4, 10
```

```
INSTALL MYPROG.*, , 4,260, First Install Disk
```

```
INSTALL *.EX_, *.EXE, 3, 85, Second Install Disk
```

```
INSTALL *.DLL, ~SYS, 2, 120, Second Install Disk, Update changed DLLs?
```

```
INSTALL *.VB_, ~SYS\*.VBX, 4, 170, Third Install Disk, Update VBXs?
```

Each of these and many more options are detailed below.

Keywords may be in upper or lower case. Some keywords have a short and a long form, and either may be used interchangeably. Keywords that require additional parameters must be followed by a space before any parameters.

Parameters may be in upper or lower case. Parameters are separated from each other by commas. Any number of spaces may also be added between parameters for readability. Parameters which themselves contain commas must be enclosed in double quotes.

Numeric parameters may be entered in either decimal or hexadecimal format. A hexadecimal number is identified by a trailing "h" or "H", as in "c000h" or "FFh" or "10h". All numeric parameters must be integers.

In most cases, keywords may be entered in any order in the settings file. Keywords that allow multiple entries, such as INSTALL, DEL, MKDIR, AUTO, CONF, REG etc will be processed in the order in which they appear in the file. The only keywords that are sequence dependent are IFILE and ISECT which must come before their corresponding INI records,

Blank lines and REMarks lines may be inserted anywhere into the settings file for readability.

Use of Variables

Many of the keywords in the settings file allow the use of variables to insert information such as file names, directory names, or other data that is not defined until the program is run.

There are three types of variables:

1. User-entered variables contain data entered by the user in response to INPUT keywords in the settings file. Up to ten user variables are available, named ~0 through ~9.
2. Program-defined variables contain information derived from the names of files, directories and other data used by K-INSTALL when it is run:

~BOOT	the boot directory
~CMD	the command to run the program derived from RUN keyword
~CPU	the name of the required processor defined by the CPU keyword
~DOC	the main documentation file defined by the DOC keyword
~DOS	the DOS directory
~DOSVER	the minimum DOS version defined by the DOSVER keyword
~WINVER	the minimum DOS version defined by the WINVER keyword
~GROUP	the Program Manager group defined by the GROUP keyword
~INST	the directory from which K-INSTALL is being run
~MAIN	the main program directory entered by the user
~NAME	the name of the installation program itself
~PROG	the name of the program defined by the RUN keyword
~SYS	the Windows System directory
~TITLE	the program title defined by the TITLE keyword
~WIN	the Windows directory

3. System defined variables contain information about the machine on which K-INSTALL is being run:

~\$CPU	the name of the processor
~\$DOSVER	the DOS version
~\$WINVER	the Windows version

All the above variables can be used by many keywords such as PATH, AUTO, SHELL, CONF, RFILE, REG, IFILE, INI, GROUP, ITEM, CATALOG, HELP, FIRST, LAST, INSTALL, LHA, ZIP, ARJ, EXE and ARCHIVE. Note that variable names must always be entered in upper case.

Getting Started Fast

K-INSTALL has many features, but you don't have to know about them all to use the program effectively - a very simple settings file will get you started. K-INSTALL uses sensible defaults for anything not specified, while still utilising many of its powerful functions. In fact, K-INSTALL can run without a settings file at all!

However, to give your program just that little customisation that you want, create a simple settings file for it. K-INSTALL is versatile enough to actually create a basic settings file for you!

1. Run INSTALL or SETUP with the special MAKE.SET settings file, for example:

INSTALL MAKE

2. Press Enter to start the "installation" process, then enter the name of the disk and/or directory that contains the program you are getting ready to distribute.
3. Answer each question with information about your program. The final question asks for the name to use for the installation program itself, such as "install" or "setup" or something else. K-INSTALL will copy itself and create a basic settings file with this name into the directory you have chosen at the start.
4. Now run INSTALL or SETUP in your directory. You already have a complete working installation program, attractively presented, that will copy everything from your disk or directory to wherever the user specifies, and includes checking the available space on the destination drive, giving a full status display as the install proceeds, automatically unpacking archives that contain an underline character, displaying information to the user at the end of the program, and allowing the user to uninstall either immediately or any time in the future.
5. Using any text editor you can continue to customise your settings file as much as you like for additional options such as multi disk installs, optional component installs, system configuration updating, your own screen layout and colour scheme, and many others. INSSAMPL.SET (for the DOS version) and SETSAMPL.SET (for the Windows version) provide a complete list of keywords for you to choose from and complete. The other sample files show different combinations of these settings.

Differences between DOS and Windows versions

The substantial functions of both the DOS and Windows version of K-INSTALL are exactly the same. The only differences are those that relate specifically to the operating environment in which K-INSTALL is running, and relate to the appearance of K-INSTALL, and the means of running the installed program.

Regarding K-INSTALL's appearance, the syntax of the WINDOW, COLS and PICTURE keywords are different, and the Windows version also has additional keywords ICON and PATTERN to take advantage of the extra capabilities of that environment.

Regarding the running of the installed program, the format of the GROUP keyword is different and the Windows version also has the ITEM keyword.

Apart from these differences, any settings file can be used interchangeably in both the DOS and Windows version.

DICTIONARY OF SETTINGS

The following list describes all available settings.

Parameters in square brackets [] are optional.

REM

Allows additional remarks to be inserted into the settings file without altering the operation of the INSTALL program.

Usage: REM [text]

text is any explanatory notes or additional information for anyone editing the settings file.

Example: REM This file last modified by Rob McDonnell 1.2.95

TITLE

Defines a descriptive name for the program to be installed, which is used throughout K-INSTALL as a title for the program.

Usage: TITLE text

text is any text including spaces.

Example: TITLE Super Space Blasters II

Default: The general term "Program" is used.

MSG

Specifies a fixed message to be displayed on the K-INSTALL screen.

Usage: MSG [text]

text is the message to be displayed.

DOS version only: text may contain colour codes to change foreground and background colours anywhere in the message. See the PICTURE keyword below for an explanation of colour control characters.

Example: MSG Another quality product from Blasterware

Default: No message is displayed.

CPU

Defines the minimum processor required by the program being installed. If the user's CPU is not greater than or equal to this value then installation is terminated.

Usage: CPU [type]

type is the type of processor:

- 1 PC1/XT
- 2 AT/286
- 3 386/486/Pentium

Examples: CPU 3

Default: No CPU defined, no checking done.

COPR

Defines whether a maths coprocessor is required by the program being installed. If the user's machine does not have a coprocessor then installation is terminated.

Usage: COPR [type]

type is whether a coprocessor is required:

- 0 not required
- 1 required

Examples: COPR 1

Default: No coprocessor specified, no checking done.

DOSVER

Defines the minimum DOS version required by the program being installed. If the user's DOS version is not greater than or equal to this value then installation is terminated.

Usage: DOSVER [ver]

ver is the version number

Examples: DOSVER 3.10

DOSVER 4

Default: No DOS version defined, no checking done.

WINVER

Defines the minimum Windows version required by the program being installed. If the user's Windows version is not greater than or equal to this value then installation is terminated.

Usage: WINVER [ver]

ver is the version number

Example: WINVER 3.1

Default: No Windows version defined, no checking done.

Note: In the DOS edition of K-INSTALL the installed Windows version is always 0.

DIR

Defines the default directory in which the program is to be installed.

Usage: DIR [dest[,opt[,ask]]]

dest is the full drive and directory name, or blank to install to the current directory.

opt is whether the user has the option of changing the directory:

0 the user is asked for a directory and can change from the default

1 the user is not asked, the dest is used without alteration

ask is the question used to ask the user to enter a directory name.

Examples: DIR C:\GAMES\BLASTER

DIR D:\ACCOUNTS, , Enter the Accounts program directory name

DIR C:\SETDIR, 1

Default: ask defaults to a standard question, opt defaults to 0, dest defaults to blank (the current directory).

INSTALL

Specifies a file or group of files to be copied and/or unpacked. It includes information about how and where to copy the files, the space required, the prompt for inserting a new disk if required, a prompt to ask the user whether to install the files at all, an additional description of the files, and a program to run when the files are installed.

Usage: INSTALL fspec[,dest[,repl[,size[,disk[,ask[,desc[,prog[,parm]]]]]]]]]

fspec is the name and extension of the file(s) to be installed. Wildcards are allowed.

dest is the directory to install the files to and/or new name for the files being copied. The directory is specified relative to the main directory entered by the user:

. is the main directory (same as blank)

.. is the parent directory of that drive

\ is the root directory of that drive

ABC is a subdirectory of the main directory

A new fspec can be added to the dest parameter to rename the files as they are copied. If there is no fspec, or there are wildcards ("*" or "?") for both the name and extension, then the file retains its original name. If there is a wildcard name and specific extension then just the extension is changed. If there is a wildcard extension and specific name then just the name is changed. If the file being copied is an archive to be automatically unpacked, then the new fspec is ignored and the file names are obtained from within the archive.

Any tilde variables may also be used - see the section "Use of Variables" above for more information.

repl is the method of deciding whether to overwrite/replace or append existing files with the same name in the destination directory:

0 don't replace anything, copy only new files

1 replace older files on user confirmation

- 2 replace older files
- 3 replace all files on user confirmation
- 4 replace all files
- 5 append to files

size is the space in KB required to install the files.

disk is the name by which to ask the user to insert another disk if no files matching the fspec are found on the current disk. This enables single or multi disk installs using the same settings file - the additional disks are only requested if necessary.

ask is the question to ask the user as to whether to install the files represented by this INSTALL record. If this is not blank then the INSTALL record is taken as a conditional install, that is, the user is asked whether they want to install them or not. The question should be asked in such a way that a positive response indicates that the files should be installed. This allows the user to choose how much of the function of the total package they wish to install. If consecutive INSTALL records contain the same question then the user is asked only once.

desc is the description of the files, such as "Help System" or "Sample Databases" which is displayed along with the file name as it is installed.

prog is the name of a program to run following installation of the files on this INSTALL record.

parm is any parameters required by the above program.

Examples: INSTALL BLAST.EXE, ., 2, 125, Install Disk 1

INSTALL BLAST.OV_*.OVL, 2, 220, Install Disk 1

INSTALL SAMPLE.*, ., 2, 140, Install Disk 2

INSTALL *.HLP, ., 1, 130, ,Do you want to install the Help files?

INSTALL SAMPLS.ZIP, SAMPLES, 2, 90, Disk 2, Install samples?

INSTALL *.DLL, ~SYS, 1, 430, 3rd Install Disk, Update DLLs?

INSTALL *.DL_~SYS*.DLL, 1, 90, 4th Install Disk, Update DLLs?

INSTALL *.EX_~MAIN*.EXE, 1, 90, , , Program files

INSTALL *.DBF, , 4, 287, , , Sample databases, convert.exe, /all

INSTALL *.DL_~SYS*.DLL, 1, 90, , , Windows DLLs

Default: prompt defaults to blank (files are installed without asking the user), disk defaults to blank (presuming a single disk install), size defaults to 0 (space on target disk will not be checked), repl defaults to 0 (copies only if files do not exist), dest defaults to blank (the main program directory entered by the user, file is not renamed). If no INSTALL records at all are entered then K-INSTALL defaults to a simple all-on-one-disk installation, ie fspec is "*.*", dest is blank, repl is 0, size is 0, disk and prompt are blank.

FIRST

Specifies a program or command to be executed at the beginning of the installation. This can be used to perform a DOS command or run some other specialised program to perform additional processing.

Usage: FIRST [cmd[,parms]]

cmd is the program or command to be executed.

parms is any additional parameters for the program.

Any tilde variables may be used in any of the above parameters - see the section "Use of Variables" above for more information.

Examples: FIRST DOFIRST.COM

FIRST MKDIR, C:\TEMP

Default: No additional processing is performed before installation.

LAST

Specifies a program or command to be executed at the end of the installation. This can be used to perform a DOS command or run some other specialised program to perform additional processing.

Usage: LAST [cmd[,parms]]

cmd is the program or command to be executed.

parms is any additional parameters for the program.

Any tilde variables may be used in any of the above parameters - see the section "Use of Variables" above for more information.

Examples: LAST DOLAST.EXE

LAST ARJ.EXE, SAMPLES

LAST PRINT, READYREF.TXT

Default: No additional processing is performed after installation.

UNPAK

Defines the identifier that indicates that an archive file is to be automatically unpacked by K-INSTALL.

Usage: UNPAK string

string is between 1 and 8 valid filename characters.

Example: UNPAK &&

Default: A single underline indicates an archive to be unpacked, so files such as _ABC.ZIP or MY_FILE.LZH ARCHIVE_EXE will be unpacked automatically by K-INSTALL into the directory defined in the relevant INSTALL record.

LHA

Sets the options used by LHA.EXE when automatically unpacking .LZH files.

Usage: LHA [parms]

parms is all the options required by the unarchiving program.

Any tilde variables may be used in the above parameter - see the section "Use of Variables" above for more information.

Example: LHA e /m1

Default: The options "e /ma1" are used.

ZIP

Sets the options used by PKUNZIP.EXE when automatically unpacking .ZIP files.

Usage: ZIP [parms]

parms is all the options required by the unarchiving program.

Any tilde variables may be used in the above parameter - see the section "Use of Variables" above for more information.

Example: ZIP -o

Default: The options "-o -Jhrs" are used.

ARJ

Sets the options used by ARJ.EXE when automatically unpacking .ARJ files.

Usage: ARJ [parms]

parms is all the options required by the unarchiving program.

Any tilde variables may be used in the above parameter - see the section "Use of Variables" above for more information.

Example: ARJ e

Default: The options "e -y -a" are used.

EXE

Sets the options used by self unpacking archives when they are run automatically.

Usage: EXE [parms]

parms is all the options required by the unarchiving program.

Any tilde variables may be used in the above parameter - see the section "Use of Variables" above for more information.

Example: EXE /s

Default: No options are used.

ARCHIVE

Defines an additional unarchiving program and associated file extension and optional parameters.

Usage: ARCHIVE ext[,prog[,parms]

ext is the file extension of files to be unarchived.

prog is the program name and extension of the unarchiving program.

parms is all the options required by the unarchiving program.

Any tilde variables may be used in any of the above parameters - see the section "Use of Variables" above for more information.

Example: ARCHIVE .ZIP, UNZIP.EXE, -o

Default: No unarchiving program defined, function not enabled.

DEL

Specifies the name of a file to be deleted prior to the commencement of installation. This can be used to remove existing files, usually from an earlier version of the same program, which are incompatible with this version. Wildcards are not permitted.

Usage: DEL path

path is the directory, file name and extension of the file to be deleted. The directory is specified relative to the main program directory entered by the user. This can include the DOS directory symbols ".", "..", "\" etc, and/or any tilde variables - see the INSTALL keyword and the section "Use of Variables" above for more information.

Example: DEL PREVIOUS.DOC

DEL ~DOS\OLD.EXE

DEL ~MAIN\UNWANTED.DAT

Default: If the directory is blank then the file is deleted from the main program directory. If there is no DEL record then no files are deleted.

MKDIR

Specifies the name of a directory to be created following completion of installation.

Usage: MKDIR dir

dir is the directory to create. The directory is specified relative to the main program directory entered by the user. This can include the DOS directory symbols ".", "..", "\" etc, and/or any tilde variables - see the INSTALL keyword and the section "Use of Variables" above for more information.

Example: MKDIR PRIVATE

MKDIR ~BOOT\TEMP

Default: No directories are created except those to which files are copied.

INPUT

Specifies a variable that may either be a constant or a user entered item. The value of this variable may then be inserted into the system configuration files AUTOEXEC.BAT and CONFIG.SYS, INI files or registration data file

using "~0" for the first value, "~1" for the second value etc. Up to 10 INPUT records may be specified, giving variables from "~0" to "~9".

Usage: INPUT num[,size[,data[,fmt[,name[,ask]]]]]

num is the variable number from 0 to 9

size is the maximum length of the variable.

data is the default value for the variable.

fmt is the format of the data to be entered. It may contain any combination of literal characters and wildcards, and relative and boolean operators.

Wildcards are similar to those in DOS, but with additional capabilities. There are four wildcard characters:

@ is for any one alphabetic character

is for any one numeric character

? is for any one character

* is for any number of characters

Use a single asterisk "*" to indicate that anything can be entered, including a blank line. Use a combination "?*" to indicate that at least one character must be entered (ie, anything except a blank). Use other wildcard combinations to restrict the valid input even further, such as "?* ?*" to indicate that the input must be at least two words, or "@@?" to indicate that the input must be three characters long, the first two of which must be alphabetic, or "#A" to indicate that the input must be a number followed by the letter "A". This facility is very useful to ensure that the user is properly registered and knows how to enter the right information in the right place.

Relative operators allow ranges to be specified. There are seven:

= is equal to (optional, this is the default)

◇ is not equal to

== is exactly equal to, wildcards treated as literal characters

> is greater than

< is less than

>= is greater than or equal to

<= is less than or equal to

Boolean operators allow negative matches or combinations. There are three:

NOT true if the criteria is not satisfied

AND true if both criteria are satisfied

OR true if either criteria are satisfied

name is the name of the data item that the user is asked to enter. If this is blank then the user is not prompted to enter a value.

ask is the question to ask the user so that they will enter the correct information.

Examples: INPUT 1,6, , @#####, Serial Number, Enter program serial number

INPUT 2,30, , ?* ?*, First and Last name

INPUT 3,60, , *, Address

INPUT 4,5, 2400, 2400 or 4800 or 9600 or 14400, Modem Speed

INPUT 5,8,, ##/##/## and >=840101, Date, Enter Date as yymmdd

Note: Ensure that the size and format are compatible. Specifying a length of 2 and a format of "???" will block the user from ever entering a valid value for the variable.

Default: If ask is blank then it is derived from name, name defaults to blank (no user input requested), fmt defaults to "*" (any input allowed), data defaults to blank (no value assigned), and size defaults to 0 (no specific size) if not entered.

RFILE

Specifies the location, name and format of a registration data file. The information in this file can be constructed from any combination of predefined and user entered data such as name, address, program serial number etc. The user data is controlled through INPUT keywords.

Usage: RFILE [path[,fmt]]

path is the full directory, name and extension of the registration data file. The directory is specified relative to the main program directory entered by the user. This can include the DOS directory symbols ".", "..", "\" etc, and/or any tilde variables - see the INSTALL keyword and the section "Use of Variables" above for more information.

The directory must exist, or be the main program directory entered by the user, or it must be created through the MKDIR keyword.

fmt is the format in which the data is written to the file:

A is Appended Text format, ie normal text appended to a file

B is Binary format, ie no length byte or null terminator

C is C language format, ie null terminated string

P is Pascal format, ie length byte followed by string

T is New Text format, ie normal text replacing any existing file

In both Appended and New Text format all information is written sequentially to a normal text file such as could be created by any editor. Appended Text format will append to the end of any existing file, while New text format will completely replace any existing file with the same name. In Binary or C or Pascal format information can be placed precisely in an existing file, or a new file created.

If more than one character is entered for fmt just the first character is examined to determine whether it is A, B, C or P or T.

Examples: RFILE BLASTER.DAT, P

RFILE ~WIN\MYPROG.INI

RFILE \STARTUP.BAT, TEXT

Default: If path is blank or there is no RFILE record, then no registration data file is defined, option not enabled. If fmt is blank or anything starting with other than "A", "B", "C" or "P" then New Text format is presumed. If there

is no directory specified then the main program directory is presumed.

REG

Specifies text to be written to a registration data file. The data may include any variables or special directory identifiers.

Usage: REG [pos[,size[,text]]]

pos is the incremental position in the file where the data is to be written in a C or Pascal or Binary format file, that is, the number of bytes to skip over from the end of the previously written text. This is ignored in a Text format file.

size is the length of the data required. In a Text format file the information is truncated to this length, in a C or Pascal or Binary format file it is truncated or padded out as necessary to exactly this length.

text is the data to be written to the file. Any tilde variables may be used - see the section "Use of Variables" above for more information.

Examples: REG 0, 0, Colour=7 4 6

REG 0, 0, Serial=~0

REG 0, 0, UserName=~1

REG 37, 16, UnlockCode872915

REG 58, 8, ~3

Default: text defaults to blank, and pos and size default to 0 if not entered. If no REG records at all are entered then K-INSTALL does not create a registration data file.

BAK

Specifies the file extension to be used when backing up AUTOEXEC.BAT, CONFIG.SYS and INI files prior to alteration.

Usage: BAK ext

ext is the file extension to be used for backups. Ensure that it commences with a full stop.

Examples: BAK .BAK

BAK .OLD

Default: A file extension of ".ARK" is used for backup files.

PATH

Specifies directories to be added to the DOS Path in AUTOEXEC.BAT. The file is updated if the directories do not already exist in the Path. Confirmation is always obtained from the user before modifying AUTOEXEC.BAT, and the file is backed up before alteration.

Usage: PATH [dirlist]

dirlist is one or more directory names separated by semicolons.

Any tilde variables may be used in the above parameter - see the section "Use of Variables" above for more information.

Examples: PATH

```
PATH ~MAIN\;~MAIN\HELP
```

Default: If dirlist is blank then just the main program directory is added to the DOS Path. If there are no PATH records then the DOS Path is not altered.

AUTO

Specifies a line that is to be added to AUTOEXEC.BAT. The line is added after all other lines in AUTOEXEC.BAT except for a final DOSHELL or WIN or other shell command (see SHELL keyword below). Confirmation is always obtained from the user before modifying AUTOEXEC.BAT, and the file is backed up before alteration.

Usage: AUTO [text]

text is the line to be added to the file. Any tilde variables may be used - see the section "Use of Variables" above for more information.

Examples: AUTO REM Run VET Anti-Virus program

```
AUTO ~MAIN\VET /#
```

```
AUTO SET VETDIR=~MAIN
```

Default: No lines are added to AUTOEXEC.BAT.

SHELL

Specifies a line that is to be added to the end of AUTOEXEC.BAT. Existing calls to DOSHELL or WIN are replaced. The PATH statement and any lines defined by AUTO keywords are inserted before these lines if necessary. Confirmation is always obtained from the user before modifying AUTOEXEC.BAT, and the file is backed up before alteration.

Usage: SHELL [text]

text is the line to be added to the file. Any tilde variables may be used - see the section "Use of Variables" above for more information.

Example: SHELL KMENU /td/vd

Default: No lines are added to the end of AUTOEXEC.BAT.

FILES

Specifies a minimum value for the FILES setting in CONFIG.SYS. The file is updated if the FILES value is less than that specified. Confirmation is always obtained from the user before modifying CONFIG.SYS, and the file is backed up before alteration.

Usage: FILES [num]

num is the minimum value for FILES that is required.

Example: FILES 20

Default: The FILES value is not altered.

BUFFERS

Specifies a minimum value for the BUFFERS setting in CONFIG.SYS. The file is updated if the BUFFERS value is less than that specified. Confirmation is always obtained from the user before modifying CONFIG.SYS, and the file is backed up before alteration.

Usage: BUFFERS [num]

num is the minimum value for BUFFERS that is required.

Example: BUFFERS 20

Default: The BUFFERS value is not altered.

CONF

Specifies a line that is to be added to CONFIG.SYS. The line is added at the end of the file. The FILES and/or BUFFERS statements are inserted before this line if necessary. Confirmation is always obtained from the user before modifying CONFIG.SYS, and the file is backed up before alteration.

Usage: CONF [text]

text is the line to be added to the file. Any tilde variables may be used - see the section "Use of Variables" above for more information.

Example: CONF DEVICE=~DOS\MSCDEX.SYS

CONF DEVICE=~MAIN\TDRIIVER.SYS /F:200 /D /L::~BOOT

Default: No lines are added to CONFIG.SYS.

IFILE

Specifies the location and name of an INI file to be created or modified. The information in this file can be constructed from any combination of predefined and user entered data such as name, address, program serial number etc. The user data is controlled through INPUT keywords. The actual data written to the file is defined with ISECT and INI keywords.

Usage: IFILE [path]

path is the full directory, name and extension of the INI file.

The directory is specified relative to the main program directory entered by the user. This can include the DOS directory symbols ".", "..", "\" etc, and/or any tilde variables - see the INSTALL keyword and the section "Use of Variables" above for more information.

The directory must exist, or be the main program directory entered by the user, or it must be created through the MKDIR keyword.

Examples: IFILE MYPROG.INI

IFILE ~WIN\WIN.INI

IFILE

Default: If path is blank, or there is no IFILE record, then data is written to WIN.INI in the WINDOWS directory. If there is a file name and extension but no directory then the main program directory is presumed.

ISECT

Specifies the section name under which the INI data is to be written. Sections are recorded in the INI file enclosed in square brackets, however the section name here should be entered without square brackets. The section will be created in the INI file if it does not already exist.

Usage: ISECT sect

sect is the name of the section to which the data is written.

Examples: ISECT Desktop

ISECT MyApplicationSettings

Default: None.

INI

Specifies data to be written to an INI file. They may be preceded and interspersed by as many IFILE and ISECT records as required to write data to multiple files and sections. The data may include any variables or special directory identifiers.

Usage: INI key[=data]

key is the name of the item for which a value is being defined, such as "Colour" or "Directory" or "Font".

data is the value being assigned to the above item, such as "Red" or "~MAIN" or "Arial". Any tilde variables may be used - see the section "Use of Variables" above for more information.

Examples: IFILE MYPROG.INI

ISECT Main

INI Colour=7 4 6

INI Dir=~MAIN

INI Serial=~0

INI UserName=~1

ISECT Defaults

INI OpenEmpty=Yes

INI AutoSave=Yes

INI PromptOnDelete=No

Default: if data is blank then any existing value assigned to that item name is deleted.

GROUP (DOS version only)

Specifies the file name of a pre-created Group to be added to Program Manager if it does not already exist.

Usage: GROUP fileext

fileext is the name of the group file. Any tilde variables may be used - see the section "Use of Variables" above for more information.

Example: GROUP MYPROG.GRP

Notes: The group file must be on the last disk in a multi disk install, and it is automatically copied into the Windows directory on the user's machine. It should not therefore be included in any INSTALL keyword.

When creating the original group file, care must be taken to ensure that it will work satisfactorily on the user's machine, which may be operating at a different screen size and resolution, and with the Program Manager window itself of a different size. Also, user-entered directories and other variables cannot be added to the group file at installation file.

Default: if there is no GROUP record specified then Program Manager is not updated.

GROUP (Windows version only)

Specifies the name of a Group to be added to Program Manager if it does not already exist.

Usage: GROUP text

text is the name of the group. Any tilde variables may be used - see the section "Use of Variables" above for more information.

Examples: GROUP Space Blaster

GROUP My Favourite Programs

Default: if there is no GROUP record specified then any Items will be added to the currently selected group.

ITEM (Windows version only)

Specifies the name and other information about a program item to be added to the Program Manager group defined with the GROUP keyword.

Usage: ITEM cmd[,name[,icon[,index[,x,y[,dir[,key[,min]]]]]]]]]]

cmd is the full command line required to run the program. This parameter should contain at least the name of the executable file for the program. It can also include the path of the file and any parameters required by it.

name is the title that is displayed below the icon in the group window.

icon is the full path for the icon to be shown in the group window. The file can be either a Windows executable file or an icon file.

index is the index of the icon in the file identified by the icon parameter. This parameter is an integer, and the first icon's index is 0. SETUP.EXE contains two icons, the first for install and the second for uninstall. PROGMAN.EXE contains five icons that can be used for non-Windows programs.

x,y is the horizontal and vertical position of the icon in the group window. These parameters are integers, and both must be specified to set the position of the icon.

dir is the name of the default (or working) directory.

key is the hot (or shortcut) key that is specified by the user.

minis whether an application window should be minimised when it is first displayed.

Notes: Any tilde variables may be used in any of the above parameters - see the section "Use of Variables" above for more information.

Any parameters which contain square brackets [] or parenthesis () must be enclosed in double quotes or Windows will generate errors and may even crash.

Examples: GROUP Space Blasters

```
ITEM ~MAIN\BLASTER.EXE,"Space Blasters Game (for DOS)"
```

```
ITEM ~MAIN\BLASTER.HLP,Help for Blasters,BLASTHLP.ICO
```

```
ITEM ~MAIN\SETUP.EXE,Uninstall Space Blasters,,1
```

Default: If min is blank then the program is not minimised when run. If hotkey is blank then no shortcut key is defined. If dir is blank then the working directory is taken from the cmd parameter. If x,y is blank then Program Manager places the icon in the next available space. If index is blank then the first icon in the icon file is used. If icon is blank then Program Manager uses the first icon in the file specified in the cmd parameter, or if that is not an executable file then the first icon in the associated executable file, or if there is no associated file then a default icon is used.

HELP

Specifies the file name and extension of a help utility that can be run from K-INSTALL.

Usage: HELP [fileext[,parms]]

fileext is the name of the help utility, or blank to turn the option off. No directory need be specified as K-INSTALL will search the installation disk, the main program directory, and the user's Path to find the file.

parms is any additional parameters for the program.

Any tilde variables may be used in any of the above parameters - see the section "Use of Variables" above for more information.

Examples: HELP README.EXE

```
HELP BROWSE.COM, README.TXT
```

Default: No help utility defined, option not enabled.

CATALOG

Specifies the file name and extension of a product catalogue program that can be run from K-INSTALL.

Usage: CATALOG [fileext[,parms]]

fileext is the name of the catalogue program, or blank to turn the option off. No directory need be specified as K-INSTALL will search the installation disk, the main program directory, and the user's Path to find the file.

parms is any additional parameters for the program.

Any tilde variables may be used in any of the above parameters - see the section "Use of Variables" above for more information.

Examples: CATALOG CATALOG.EXE

CATALOG BROWSE.COM, PROGRAMS.TXT

Default: No catalogue program defined, option not enabled.

RUN

Specifies the file name and parameters by which to run the main program being installed. This allows K-INSTALL to give instructions on how to run the program, and also allows the user to run it directly from the installation program. This keyword can be omitted if the program is a collection of utilities or text files.

Usage: RUN [fileext[,parms[,opt]]]

fileext is the file name and extension of the program to be run. No directory need be specified as K-INSTALL will search the installation disk, the main program directory, and the user's Path to find the file.

parms is any additional parameters for the program.

opt is when and how the program should be run (DOS version only):

0 run immediately

1 exit from K-INSTALL and then run

2 exit from K-INSTALL, load Windows and then run

If opt is 0 then when the program is terminated control is returned to K-INSTALL, but if opt is 1 or 2 then control returns to DOS or from wherever K-INSTALL was run.

Any tilde variables may be used in any of the above parameters - see the section "Use of Variables" above for more information.

Examples: RUN CLOCK.COM

RUN BLAST.EXE, /VGA /NOSOUND

Note: Specifying opt of 1 or 2 forces K-INSTALL to load the program name (minus extension) and parameters into the keyboard buffer before terminating. If opt is 2 then the "win" command is loaded into the keyboard buffer first.

One disadvantage of this method is that it probably won't work if K-INSTALL is itself run from a shell or from Windows. Another disadvantage is that there is limited space in the keyboard buffer (usually 16 characters) to store keystrokes, so lengthy parameters cannot be specified as they will not fit into the buffer. If the program has lengthy parameters then put the program name and its parameters in a batch file and run that instead from K-INSTALL.

RESTART (Windows version only)

Specifies the conditions under which the user is given the option to restart Windows after installation.

Usage: RESTART [level]

level is the criteria for restarting Windows:

0 never restart Windows

1 restart Windows if WIN.INI or SYSTEM.INI have changed

2 always restart Windows

Example: RESTART 2

Defaults: 0 if level is blank, 1 if there is no RESTART keyword

REBOOT

Specifies the conditions under which the user is given the option to reboot DOS after installation.

Usage: REBOOT [level]

level is the criteria for rebooting DOS:

0 never reboot DOS

1 reboot DOS if AUTOEXEC.BAT or CONFIG.SYS have changed

2 always reboot DOS

Example: REBOOT 0

Defaults: 0 if level is blank, 1 if there is no REBOOT keyword

DOC

Specifies the name of a README or other information file or document to which the user's attention should be drawn at the termination of K-INSTALL.

Usage: DOC [filext]

filext is the file name and extension of the documentation file. No directory should be specified, as this file name is used for reference only.

Examples: DOC BLAST.TXT

DOC README.NOW

Default: No documentation file defined, option not enabled.

COLS (DOS version only)

Specifies the colours used throughout K-INSTALL.

Usage: COLS [titl[,msg[,pic[,btn[,menu[,stat[,ask[,norm]]]]]]]]]]]

titl is the absolute colour for the title line.

msg is the absolute colour for the message line.

pic is the absolute colour for the picture to start with.

btn is the absolute colour for the push buttons.

menu is the absolute colour for the main menu.

- stat is the absolute colour for the status bar.
- ask is the absolute colour for the questions.
- norm is the absolute colour for other program responses.

Notes: Absolute colour codes specify a complete foreground and background colour combination, and are defined in a single byte, giving a decimal number from 0 to 255, or a hexadecimal number from 00h to FFh. The foreground colour is defined in the lowest 3 bits, the intensity in the next bit, the background colour in the next 3 bits, and blinking in the highest bit. There are eight basic colours for both background and foreground, but the foreground colours can also be selected in high or low intensity, effectively giving 16 foreground colours.

Val	Low Intensity	High Intensity
0	Black	Dark Grey
1	Blue	Light Blue
2	Green	Light Green
3	Cyan	Light Cyan
4	Red	Light Red
5	Magenta	Light Magenta
6	Brown	Yellow
7	Light Grey	White

To derive the colour code for a specific absolute colour: take the basic colour number for the foreground, add 8 if you want high intensity, add the basic number for the background colour multiplied by 16, and add a further 128 if you want blinking.

Examples: COLS 7, 7, 7, 7, 7, 7, 7, 7

COLS 79

Default: all colours default to reasonable values if not entered, and remain unchanged if the colour is set to zero.
Defaults are:

title line	White on Blue
message line	Light Green on Green
picture	White on Black
push buttons	Dark Grey on Light Grey
main menu	Yellow on Green
status bar	Yellow on Blue
questions	Red on Black
other responses	Cyan on Black

COLS (Windows version only)

Specifies the colours used throughout K-INSTALL.

Usage: COLS [top[,bot[,titl[,shad[,depth[,msg[,menu]]]]]]]]]

top is the background colour at the top of the window.

bot is the background colour at the bottom of the window.

titl is the text colour for the title line.

shad is the text colour for the title shadow.

depth is the amount of offset of the shadow from the title line, where positive numbers are below and to the right, and negative numbers are above and to the left, and zero means no shadow.

msg is the text colour for the message line.

menu is the text colour for the main menu items.

Notes: Colours are made up three components, Red, Green and Blue. Each component has an intensity from 0 to 255, specified in one byte. A complete colour code therefore has three bytes, giving decimal numbers from 0 to 16777215, or hexadecimal from 000000h to FFFFFFFh. The lowest order byte is the Red value, the next byte is the Green value, and the highest byte is the Blue value. Any number of intermediate colours can be made by mixing different intensities of these three colours.

To derive the colour code for a specific colour: take the value for Red, add the value for Green multiplied by 256, and add the value for Blue multiplied by 65538. In hexadecimal, write the Blue byte followed by the Green byte followed by the Red byte. Using codes where each of the three colours are divisible by 64 (40h) minimises dithering. The most common and attractive colour codes are:

Decimal	Hex	Colour
0	000000h	Black
128	000080h	Dark Red
255	0000FFh	Light Red
32768	008000h	Dark Green
32896	008080h	Brown
65280	00FF00h	Light Green
65535	00FFFFh	Yellow
8288608	800000h	Dark Blue
8388736	800080h	Dark Magenta
8421376	808000h	Dark Cyan
8421504	808080h	Dark Grey
12632256	C0C0C0h	Light Grey
16711680	FF0000h	Light Blue
16711935	FF00FFh	Light Magenta
16776960	FFFF00h	Light Cyan
16777215	FFFFFFh	White

The depth quality normally works best with just small values up to about 3 and down to about -3.

Examples: COLS 255, 0, 0, 16777215, 0, 2, 16777215, 12632256

COLS ff0000h ff00ffh, 2, 0, fffffh, 1

Default: all colours and options default to reasonable values if there is no COLS keyword:

top background	Light Cyan
bottom background	Light Green
title	White
shadow	Black
shadow depth	1
message	Black
main menu	Black

Individual colours default to black and depth defaults to zero if they are missing from the COLS record.

WINDOW (DOS version only)

Defines the position and size of the main K-INSTALL window.

Usage: WINDOW x[,y[,w[,h]]]

x is the horizontal location of the window in characters from the left of the screen.
y is the vertical location of the window in lines from the top of the screen.
w is the horizontal size (width) of the window in characters.
h is the vertical size (height) of the window in lines.

Example: WINDOW 0, 0, 80, 13

Default: x and y default to zero, w defaults to 80, h defaults to 11 if there is no WINDOW record.

WINDOW (Windows version only)

Defines the position and size of the main K-INSTALL window.

Usage: WINDOW [max[,chg[,x[,y[,w[,h]]]]]]

max is the initial state of the window:

0 restored

1 maximised

chg is whether the window is resizeable and in what way:

0 the window is fully resizeable, it has a thick border for resizing, and maximise and minimise

buttons

1 the window can be maximised, restored and minimised only, it has maximise/restore and minimise buttons

2 the window can be minimised only, it has a minimise button

3 the window is not resizable at all.

x is the horizontal location of the window in pixels from the left of the screen.

y is the vertical location of the window in pixels from the top of the screen.

w is the horizontal size (width) of the window in pixels.

h is the vertical size (height) of the window in pixels.

Examples: WINDOW 1, 2

WINDOW 0, 0, 20, 20, 650, 400

Default: max, chg, x and y default to zero, w defaults to 600, h defaults to 370 if there is no WINDOW record.

PATTERN (Windows version only)

Defines the makeup of the background of the main K-INSTALL screen. The background can be either a small bitmap repeated over the whole screen, or a hatched pattern, or two plain colours that merge into each other in the middle of the screen. Only one of these three methods is used to generate the background, and K-INSTALL will attempt each in turn until valid parameters are found. If a bitmap is used then it must be added to the EXE file using a resource editor, or exist as a separate BMP file.

Usage: PATTERN [name[,style[,skew]]]

name is the name of a bitmap resource or file name. The bitmap may be up to 8 bits square, and is painted repeatedly all over the background of the window. If the bitmap is larger than 8 bits square then only the first 8 bits in each direction are used. If name is valid then the style and skew parameters are ignored.

style is the type of hatch that used to form a background pattern:

0 no hatch

1 horizontal lines

2 vertical lines

3 forwards diagonal lines

4 backwards diagonal lines

5 vertical and horizontal crossed lines

6 forwards and backwards diagonal crossed lines

This parameter is only used if the name parameter is blank or invalid. If style is greater than 0 then the skew parameter is ignored. The foreground colour of the hatch pattern is the top colour as defined by the COLS keyword. The background colour of the hatch pattern is the bottom colour as defined by the COLS keyword.

skew is the relative weight given to the top and bottom background colours in the way in which they graduate from one to the other. Positive numbers give more emphasis to the top colour, and negative numbers give more emphasis to the bottom colour, and zero means both colours are weighted evenly. The top and bottom colours are

defined by the COLS keyword. This parameter is only used if the name parameter is blank or invalid and the style parameter is blank or less than or equal to 0.

Notes: Normally only one of the above three parameters need be entered, as the first valid non-blank/non zero parameter will define how the background is generated.

Examples: PATTERN MINILOGO.BMP

PATTERN BITMAP_1

PATTERN , 6

PATTERN , , 1

Default: name defaults to blank and style and skew default to 0. If a valid name is entered then the background is generated from the bitmap and the style and skew parameters are ignored. If name is blank or invalid and style is greater than 0 then the background is generated from the hatch pattern and the skew parameter is ignored. If name is blank or invalid and style is less than 1 then the background is a graduated colour change from the top to the bottom of the screen using the skew parameter if entered. If there is no PATTERN record then the background is a simple graduated colour change using a skew of 0.

ICON (Windows version only)

Defines an icon that is displayed in the main K-INSTALL screen. The icon must be added to the EXE file using a resource editor. The EXE file already contains two icons, the first being the Install icon and the second being the Uninstall icon.

Usage: ICON [index[,x1[,y1[,x2[,y2]]]]]

index is the number of the icon resource.

x1 is the horizontal location of the icon as an absolute number of pixels from the left of the window. This is added to the value derived from x2 if defined.

y1 is the vertical location of the icon as an absolute number of pixels from the top of the window. This is added to the value derived from y2 if defined.

x2 is the horizontal location of the icon as a percentage of the way across the window, 50 being the middle and 100 the right edge. This is added to the value derived from x1 if defined.

y2 is the vertical location of the icon as a percentage of the way down the window, 50 being the middle and 100 the bottom edge. This is added to the value derived from y1 if defined.

Example: ICON 1, 0, 20, 50, 0

Default: x1, y1, x2 and y2 default to zero if not entered. If index is blank or invalid or there is no ICON record then no icon is defined or displayed.

PICTURE (DOS version only)

Defines the picture that is displayed to the top right of the K-INSTALL screen. This picture has a fixed height of 10 lines and a default width of 21 characters. This MUST be the last keyword used in the settings file, because once this keyword is encountered all of the rest of the file is read in to define the picture.

Usage: PICTURE [width]

[string

[string
[string...]]]

width is the number of characters width of the picture.

string is ASCII characters and colour control codes which make up each line of the picture on the screen. Any ASCII characters greater than or equal to 8 may be used to create the picture. ASCII characters from 1 to 7 are used to alter the colours in the picture, as follows:

- #1 set colour to the code of the following character
- #2 toggle between high and low intensity (xor 8)
- #3 shift the foreground down one colour (subtract 1)
- #4 shift the foreground up one colour (add 1)
- #5 shift the background down one colour (subtract 16)
- #6 shift the background up one colour (add 16)
- #7 toggle between blinking and non blinking (xor 128)

The #1 character must be followed by another character which specifies the colour to switch to, the ASCII value of which is the absolute colour code required.

See the COLS keyword above for an explanation of how to calculate absolute colour codes.

Example: PICTURE 0

Notes: Ensure that each line fills up the full width of the picture, putting blanks at the end of lines as necessary.

Remember that you can set an absolute colour with an ASCII character #1 followed by an ASCII character specifying the complete colour combination required, or you can set the colours relative to the last colour with the ASCII characters #2 to #7.

The pic parameter of the COLS keyword sets the starting colour of the picture. Alternatively you can simply start the picture with an ASCII #1 character and then an absolute colour character.

A very few colour combinations give absolute colour characters that may be interpreted as something else. For example to set colours to Light Magenta on a Black background you may want to use character codes #1 then #13. However the #13 will be read as a carriage return. In this case it is necessary to give a slightly different control code sequence to achieve the same result, such as #1 then #14 to get Yellow on Black, and then #3 to shift the foreground colour down one to Light Magenta.

Do not use absolute colour character #0 (Black on Black).

Default: If the width is blank or less than 0 then it defaults to 21, or if it is 0 then no picture is displayed. If no PICTURE keyword is entered at all then a simple Southern Cross is displayed.

PICTURE (Windows version only)

Defines a graphic that is displayed in the main K-INSTALL screen. The graphic is a bitmap that must be added to the EXE file using a resource editor, or exist as a separate BMP file.

Usage: PICTURE [name[,x1[,y1[,x2[,y2[,style[,x3[,y3[,x4[,y4[,mode]]]]]]]]]]]]]

name is the name of the bitmap resource or file name.

x1 is the horizontal location of the graphic as an absolute number of pixels from the left of the window. This is added to the value derived from x2 if defined.

y1 is the vertical location of the graphic as an absolute number of pixels from the top of the window. This is added to the value derived from y2 if defined.

x2 is the horizontal location of the graphic as a percentage of the way across the window, 50 being the middle and 100 the right edge. This is added to the value derived from x1 if defined.

y2 is the vertical location of the graphic as a percentage of the way down the window, 50 being the middle and 100 the bottom edge. This is added to the value derived from y1 if defined.

style is the method by which the bitmap is superimposed over the existing background of the window:

- 0 OVLY: the bitmap is shown with the background showing through the white parts
- 1 COPY: the bitmap is copied over the top of the background, completely replacing it.
- 2 AND: the bitmap and background are combined bit by bit with the AND function.
- 3 OR: the bitmap and background are combined bit by bit with the OR function.
- 4 XOR: the bitmap and background are combined bit by bit with the XOR function.
- 5-9 other effects.

The first two styles are generally the most useful, but it is worth experimenting to see the different effects they all have.

w1 is the width of the graphic as an absolute number of pixels. This is added to the value derived from w2 if defined. If the total value is not zero then the graphic is stretched or compressed to this width.

h1 is the height of the graphic as an absolute number of pixels. This is added to the value derived from h2 if defined. If the total value is not zero then the graphic is stretched or compressed to this height.

w2 is the width of the graphic as a percentage of the width of the window, 50 being half the width and 100 the whole width. This is added to the value derived from w1 if defined. If the total value is not zero then the graphic is stretched or compressed to this width.

h2 is the height of the graphic as a percentage of the height of the window, 50 being half the height and 100 the full height. This is added to the value derived from h1 if defined. If the total value is not zero then the graphic is stretched or compressed to this height.

mode is the method by which the bitmap is stretched:

- 1 preserves black pixels at the expense of white ones
- 2 preserves white pixels at the expense of black ones
- 3 treats all colours in the same way

Note: Transparent bitmaps (that is, those copied with style 0) cannot be stretched.

Example: PICTURE MAINLOGO.BMP, 0, 0, 100, 50, 1

Default: x1, y1, x2, y2, style, w1, h1, w2, h2 and mode all default to zero if not entered. If name is blank or invalid or there is no PICTURE record then no graphic is defined or displayed.

Specifies the name of an installation log file used to record all files that have been copied and directories created by K-INSTALL. This allows the program to be uninstalled by deleting all these files and directories again, regardless of where they were copied or created. The log file is always created in the main program directory.

Usage: LOG [fileext]

fileext is the file name and extension of the log file, or blank to turn the option off. No directory should be specified as the log file is always created in the main program directory.

Examples: LOG BLASTER.LOG

LOG

Default: Option enabled with log file name "UNINSTAL.LOG".

Note: For the uninstall process to work, the INSTALL program itself must be copied to the main program directory where the log file is also created. The associated settings file must also be copied if it changes the name of the log file. This can be done with a simple INSTALL record such as:

```
INSTALL INSTALL.* , , 2, 18
```

REMOVE

Specifies the name of a file to be deleted during uninstallation. This can be used to remove files that are created by the main program subsequent to installation, such as configuration files, private INI files, temporary files etc. Wildcards are not permitted.

Usage: REMOVE path

path is the directory, file name and extension of the file to be deleted. The directory is specified relative to the main program directory entered by the user. This can include the DOS directory symbols ".", "..", "\" etc, and/or any tilde variables - see the INSTALL keyword and the section "Use of Variables" above for more information.

Examples: REMOVE MYPROG.CFG

```
REMOVE ~MAIN\SORT.TMP
```

```
REMOVE ~WIN\MYPROG.INI
```

Default: If the directory is blank then the file is deleted from the main program directory. If there is no REMOVE record then no files are deleted.