

BT Install Utility - Index

[Overview](#)

[Conditions](#)

[Example](#)

[Structure of the informationfile \(script\)](#)

[Controlcodes](#)

[Create YOUR installation disk](#)

This programm **BT Install Utility V1.3** was developed and programmed by

Bruno Tille
Hördtstr. 45
70435 Stuttgart
Germany

Telephon+FAX	+49 711/873394
BTX	+49 711/873394-0001
CompuServe-ID	100113,3000

for distribution with own Windows-applications.

You are allowed to use this program free for your **private** purposes. If you use it in a **commercial** way, you have to buy a [licence](#) (outside europa available via [SWREG](#), registration ID #2316) from Bruno Tille. With this licence you are allowed to distribute your applications with the BT Install program.

You are not allowed to change copyright information or sell this program as your own product.
All rights remain with Bruno Tille.

The program is distributed "as is", you use it at your own risk.

If you have any questions, critics or suggestions, please contact me at one of the adresses above.

If you want to write your own install routine, the source code of BT Install is available for little charge. In this case please contact me.

... and now

let´s start !

Overview

BT Install is a Windows application, which allows you professional installation of other Windows applications.

Options:

- the installation process is controlled via an informationfile (.INF, also called script). In the script file it is specified, which directories should be created, which files are to copy whereto, how the programgroup should be named and which items are to put in the group.
- installation could be started from hard disk or from one of up to 9 floppy disks. If installing from floppy disks, each disk has a unique name, which is displayed at request time. For disk identification the volume label is used. The scriptfile is identical for both install methods, if you install from hard disk, no disk changes are required.
- the program directory is entered by the user at runtime. A default value is specified in the informationfile. If the directory already exists, a confirmation for overwriting is possible.
- additional to the program directory other directories may be created. Overwriting confirmation is supported.
- the windows- and the windows-system-directory are detected at runtime and are available through special control codes (&w, &s)
- program files compressed with Microsoft's COMPRESS (V2.0 or higher) are expanded automatically during installation.
- if files already exist, overwrite confirmation is supported. Additional the date/time-information of the file to install and the already existing file is compared and the result (older/newer/equal) is also displayed.
- the installation may be aborted at any time.
- BT Install creates or changes a program group and adds items to this group. The items are configurable.
- you may specify a minimal available disk space.
- BT Install writes a logfile during installation process. This logfile has the name BTInst.Log and is located in the program directory.
- it is possible to create or edit INI-files. Within the INI-files you could add new section or alter the values of existing entries.

Conditions

To use BT Install the following conditions must be satisfied:

- Windows Version 3.1 or higher, for file compression Microsoft's COMPRESS V2.0 or higher.
- you have created an informationfile (.INF, also called scriptfile) with the same name as the installationprogram. So, if your installationprogram is named INSTALL.EXE, the informationfile must be INSTALL.INF.
- the first disk must contain the informationfile and the installation program, the other disks may contain only "data".
- Windows 3.1 and the program manager must run. If the program manager is replaced with another shell, the creation of the program group may fail. However, the program files are copied.

Example

For example here is the file INSTALL.INF, which is the script for the installation of BT Install:

Install.inf

```
G-1 [General]
G-2 ProgramTitle=BT Install Utility
G-3 ProgramDir=\BTInstal
G-4 MinDiskFree=1000

D-1 [Disks]
D-2 1=Installationdisk, DISK-1

d-1 [Directories]

F-1 [Files]
F-2 1:Install.Exe=BTInst.Exe, BT Install Utility programfile
F-3 1:BTInst.Inf=BTInst.Inf, BT Install scriptfile
F-4 1:BTInst.Hl_ =BTInst.Hlp, BT Install helpfile
F-5 1:Read.Me=Read.Me, Last minute information
F-6 1:Ctl3D.Dl_=?&s\Ctl3D.Dll, Windows 3D library

I-1 [INI-Files]
I-2 1=BTInst.Ini | General | Key1 | Value1
I-3 2=BTInst.Ini | General | Key2 | Value2
I-4 3=BTInst.Ini | General | Key1 | ,Value1-2 | APPEND
I-5 3=BTInst.Ini | Special | SpecKey1 | SpecValue1

P-1 [ProgramGroup]
P-2 GroupName=BT Install
P-3 1=BTInst.Exe, BT Install, , 0
P-4 2=&nWinHelp.Exe &p\BTInst.Hlp, BT Install, , 0
P-5 3=&nNotepad.Exe &p\BTInst.Inf, BT Install Script, , 0
P-6 3=&nNotepad.Exe &p\Read.Me, BT Install Readme, , 0
P-7 5=&nNotepad.Exe &p\BTInst.Log, BT Install Log, , 0
```

Line	Explanation
G-1	section for general information
G-2	name of the program to install
G-3	default value of the program directory
G-4	required disk space in KB
D-1	section for installation disk description
D-2	name and volume label of the installation disks (here: only one)
d-1	section for additional directory creation
F-1	section for files to copy
F-2	copy Install.Exe from disk 1 to BTInst.Exe. If Install.Exe is a compressed file (with Microsoft's COMPRESS utility) it is expanded automatically. Destination is per default the program directory. The text is displayed during the copy process .
F-6	copy Ctl3D.Dl_ from disk 1 to the windows system directory. If the file already exists, the user is asked for overwrite.
I-1	section for create/edit INI-files
I-2	In BTInst.INI (located in the program dir) the key "Key1" in the section "General" is set to Value1. If BTInst.INI does not exist, it will be created. Is there no section [General], a section with this name is created. If there is already an key Key1, the old value is overwritten.
I-3	create a second key in [General]
I-4	the value of Key1 is altered. Using APPEND, the old value is not overwritten, but the nw

one appended.

After executing the [INI-Files]-commands the file BTInst.Ini looks like:

```
[General]
Key1=Value1, Value1-2
Key2=Value2
```

```
[Special]
SpecKey1=SpecValue1
```

- P-1 section for program group and icons
- P-2 name of the programgroup to be created
- P-3 create an icon labeled "BT Install" within the group, the command line is BTInst.Exe and is expanded with the program directory. The first icon of the EXE-file is taken for presentation.
- P-4 create another icon calling WinHelp.Exe. To avoid automatic expansion of WinHelp.Exe with the program directory, the controlcode &n is used.. By use of &p BTINST.HLP will be expanded by the programdirectory.
- P-7 create an icon to show the installation log file

Structure of the informationfile (Scriptfile)

The scriptfile is build like a Windows INI-file (Win.Ini, System.Ini).

You can create a scriptfile with any editor and store it in plain format.

The scriptfile is divided into sections:

Section	Contents
<u>[General]</u>	contains general information like application name, default program directory and required diskpace. This section is reequired.
<u>[Disks]</u>	Specifies the names and volume labels of the installation disks. By default, the disks are named Nr. <n> with volume label DISK-<n>. This section is optional.
<u>[Directories]</u>	Here you may enter directories to create additional to the program directory. This section is optional.
<u>[Files]</u>	List of files to be copied during installation.
<u>[INI-Files]</u>	list of INI-files to edit or create. Each entry specifies the name of the INI-file together with the section and the key to alter.
<u>[ProgramGroup]</u>	Describes the program group to be created and the items to be inserted inthis group. This section is optional.

The section [General]

This section holds general information.

ProgramTitle =<name>

Name of the application to install. The name is displayed during installation and has no further meaning.

ProgramDir =[?][d:]dir

Default for the program directory. A question mark (?) at first position causes BT Install to ask for confirmation, if the directory already exists. The user may confirm, choose a new directory or abort the installation. If no drive is specified, the drive holding the windows directory is taken.

dir may contain one or more of the controlcodes &n, &w, &s and &p.

MinDiskFree =<kb>

minimal disk space in KB required for installation. If the space is not available, a message is displayed and the program is aborted.

The section [Disks]

In this optional Section the names and volumelabels of the installation disks are specified.

For each disk place an entry like

n = <disk-name> [, <vol-label>]

with

Symbol	Meaning
n	number of the disk, starting with 1. Maximal 9 disks are allowed.
disk-name	name of the disk like "Installationdisk" or "Additional programs". If it is necessary to insert a diskette, this text is displayed in a dialog box. By default the names are "Nr. n" like "Nr. 1", "Nr. 2".
vol-label	the volumelabel of the n-th disk. If a new diskette is inserted, BT Install checks the volume label against the value specified here. By default the label is "DISK-n" like "DISK-1", "DISK-2". The fist disk (containing the installation program) is not checked.

The section [Directories]

If you want to create additional directories, place one entry for each directory like:

n = [?][d:]dir

with

Symbol	Meaning
n	entry number, starting with 1.
[?][d:]dir	name and location of the directory to create. A question mark at the first position requires a confirmation of the user, if the directory already exists. If no drive is given, the drive to where the application is installed is taken. If no absolute path is given, the new directory is local to the program directory. dir may contain one or more of the controlcodes <u>&n</u> , <u>&w</u> , <u>&s</u> and <u>&p</u> .

The section [Files]

For each file to be copied from the installation disks make an entry like:

[n:]<source> = [?][d:][[&w][&s][&p]]<dest>[, descr]

with

Symbol	Meaning
n	number of the disk, which holds the file to be copied. If no number is specified, the last disk is assumed. At installation start the disk number is set to 1.
source	name of the source file. The given name may contain path information, which is interpreted as local to the installation drive- and path. If the file is not found, the installation will be terminated displaying an informative message. If the file is packed with Microsoft's COMPRESS (2.0 or higher), it is expanded.
dest	destination of the file. A question mark (?) at the first position causes overwrite confirmation of existing files. The date/time-information of the related files (the one to install and the already existing) is compared and the user is informed, if the existing file is older, equal or newer than the file to install. The user may confirm, skip this file or abort the installation. If no drive is given, the drive to where the application is installed is taken. If no absolute path is given, the file is copied to the program directory. The specified value may contain one or more of the controlcodes <u>&n</u> , <u>&w</u> , <u>&s</u> and <u>&p</u> .
descr	file information text, displayed during copy of file.

The section [INI-Files]

This optional section controls the creation or alteration of one or more INI-files.

Each Windows INI-file is build of **Sections**, within a section there are one or more **Keys**:

```
[Section-1]
Key_1-1=Value
Key_1-2=Value
Key_1-3=Value
Key_1-4=Value
```

```
[Section-2]
Key_2-1=Value
Key_2-2=Value
```

etc...

Each INI-entry to change is described by:

n = INI_FN | Section | Key | Value [| APPEND]

with

Element	Meaning
n	number of item, starting with 1.
INI_FN	name of the INI-file. If a file with this name does not exist, it will be created. INI_FN may contain the controlcodes <u>&n</u> , <u>&w</u> , <u>&s</u> and <u>&p</u> . For example, to address the file WIN.INI, enter &w\WIN.INI .
Section	name of the section. If the section does not exist, it will be created.
Key	name of the key within section. If Key is not already existing, a new entry is created, otherwise the existing value will be overwritten (see APPEND below). If Key is empty, the whole section is removed from the INI-file.
Value	value of the key-entry. Value may contain the controlcodes <u>&n</u> , <u>&w</u> , <u>&s</u> , <u>&p</u> , <u>&d</u> and <u>&t</u>
APPEND	if you specify APPEND, already existing values for key aren't deleted. The specified value is appended at the end of the already existing. Example: in WIN.INI the section [Windows] entry [Run] should be expanded with the value MyProg.Exe, i.e. already existing entries should not be deleted. To achieve this, you should write: 1=&w\WIN.INI Windows Run MyProg.Exe APPEND

The functionality is given by the Windows API routine **WritePrivateProfileString**.

The section [ProgramGroup]

This optional sections controls the creation or altering of the applications programgroup. In order to create or alter the program group, the Windows program manager must be available.

GroupName = <name>

Name of the program group. If the group does not exist, a new group is created. In an existing group the items specified in the list below are changed, additional items remain unchanged.

For each item to be inserted write a line like

n = <cmdline>, <title> [, iconfile [, icon#]]

with

Symbol	Meaning
n	number of item starting with 1.
cmdline	command line to start the program connected with the item. Cmdline may contain one or more of the controlcodes <u>&n</u> , <u>&w</u> , <u>&s</u> and <u>&p</u> .
title	text explaining the icon (label).
iconfile	file containing the icon resource for the item. By default this is the name specified in the cmdline.
icon#	index of the icon resource within the iconfile. If iconfile is not specified, cmdline is used. Icon resource index starts with 0. If either cmdline nor iconfile contains an icon resource, a default icon is used.

Controlcodes

Controlcodes are used to complete paths or filenames. If a path/filename is entered **absolut** (for example C:\WINDOWS\XX.YY), no expansion is performed.

Relativ names are expanded by default with the program directory, if they stand alone or at the beginning of a commandline. For example, BTINST.EXE used as programname for creation of a program group icons will be expanded to C:\BTINST\BTINST.EXE. But in the line &nWINHELP.EXE BTINST.HLP the expansion mechanism must be invoked by using the controlcode &p in front of BTInst.Exe: &nWINHELP.EXE &pBTINST.EXE.

Important:The controlcodes **must** be written in **lower case letters**, otherwise the expansion is not performed !

Windows-Directory (&w)

The controlcode &w causes the expansion with the windows directory (usually C:\WINDOWS).

Windows-System-Directory (&s)

The controlcode &s causes the expansion with the windows system directory (usually C:\WINDOWS\SYSTEM).

Program-Directory (&p)

&p is replaced with the program directory. This control code is only necessary to expand file- or directorynames if used as parameters.

No directory name expansion (&n)

To avoid any expansion of a file- or directory name, set the control code &n in front of the name.

Date (&d)

&d is replaced with the date of the installation (format mm-tt-jjjj). &d may not be used in file- or directorynames !

Time (&t)

&t is replaced with the time of the installation (format hh:mm:ss). &t may not be used in file- or directorynames !

Create YOUR installationdisk

Here is an example how to create an installation disk for your application using BT Install.

1. General:

Suppose, your application consists of an EXE-file (MyApp.Exe), a help-file (MyApp.Hlp) and of a ReadMe-file (Read.Me).

During installation there should be created a subdirectory named DATA.

Also create an INI-file (MyApp.Ini). This file should hold the date and time of the installation and the complete path to the DATA-directory.

Additional insert an entry for the application-files (*.MYA) in WIN.INI under [Extensions].

The EXE- and the helpfile are compressed using Microsoft's COMPRESS.

The installationprogram itself will be named INSTALL.EXE.

2. Creating the INF-file:

Create the INF-file with a windows editor like Notepad.Exe. Each editor may be used, the only condition is to save the INF-file as plain text.

For our example the file INSTALL.INF looks like:

INSTALL.INF

```
[General]
ProgramTitle=My App - Installation
ProgramDir=\MyApp
MinDiskFree=750

[Disks]
1=Installationdisk 1, DISK-1

[Directories]
1=&p\DATA

[Files]
1:MyApp.Ex_ =MyApp.Exe, Programfile for MyApp
1:MyApp.Hl_ =MyApp.Hlp, Helpfile for MyApp
1:Read.Me=Read.Me, Last minute information

[INI-Files]
1=MyApp.Ini | Install | Date | &d
2=MyApp.Ini | Install | Time | &t
3=MyApp.Ini | Install | DateTime | &d &t
4=MyApp.Ini | Data | Dir | &p\DATA
5=&w\WIN.INI | Extensions | MYA | &p\MyApp.Exe ^.MYA

[ProgramGroup]
GroupName=My App
1=MyApp.Exe, My App, , 0
2=&nWinHelp.Exe &p\MyApp.Hlp, My App Help, , 0
3=&nNotePad.Exe &p\Read.Me, Read me, , 0
```

3. Creation of the installationdisk:

a) Compression of EXE- and HLP-files:

```
COMPRESS MyApp.Exe MyApp.Ex_
```

```
COMPRESS MyApp.Hlp MyApp.Hl_
```

you may choose other names for the compressed files. BT Install recognizes compressed files automatically.

b) Copy the files to the installation disk:

```
Copy MyApp.Ex_ A:
```

```
Copy MyApp.Hl_ A:
```

```
Copy Read.Me A:
```

Copy C:\BTInstall\BTInst.Exe A:Install.Exe
Copy Install.Inf A:

- c) Set volume label to DISK-1:
LABEL A: DISK-1 (or use other utility)
If you use only one installation disk, the volume label is not checked.

4. Test of the installation:

Test for correct execution before you give away your disk !

Registration of BT Install

You have had the chance to evaluate the goodness of this installation program.
If you want to use it furthermore in a commercial way, you must buy a licence from Dipl.Inf. B. Tille.

Outside of Germany, the best way to do so is to GO SWREG and to register the #2316.
The fee is \$60.

With this licence you are allowed to distribute your applications with the BT Install program.

