

MAINTINI V1.20

Maintini is a small Utility to modify any kind of INI Files.

Usage:

Maintini uses commandfiles (Extension CMD), which contains the modifications which should be made to the INI file.

```
MAINTINI <INI-File> <CMD-File>
```

is the Syntax to call the program. If you want to modify multiple INI files, then you can create a ICM (Ini CoMmand File). This file contains on each line a INI file and a command file.

```
MAINTINI /c <CMD-File>
```

```
;  
; Example ICM File  
;  
C:\WINDOWS\WIN.INI      C:\UT\WIN.CMD  
C:\WINDOWS\SYSTEM.INI  C:\UT\SYSTEM.CMD
```

A backup of the INI-File with the extension .BAK is made before modifying the file.

Normaly all section, keys and values are NOT case sensitive. However _ and * give the possibility to match only if the case matches too.

All lines which begins with ; are treated as comments.

MAINTINI can process INI-Files of any size, even larger than 64kBytes. Just the CMD File should not have to much commands, because a dynamic list with all commands is created. This is normaly not a problem, because you can use more than 100 commands in one CMD file. If this isn't enough, then you must splitt you CMD file and use a ICM file to do all changes.

INI Commands

;	When this is in front of a line, then this line is not used as a command. It is just ignored.
-[section]	The corresponding section is deleted
-[section]Key	The Key in section will be deleted
-[section]Key=Value	The Key is only deleted if the Value matches exactly This is used to delete device=... lines in system.ini
_[section]Key=Value	Same as - but Value is case sensitive Normaly not used.
~[section]Key=Value	Used to delete Keys, when the Key is terminated by a numeric value. (Used to delete Groups in PROGMAN.INI)
#[section]	The corresponding section is commented
#[section]Key	The Key in section will be commented

#[section]Key=Value	The Key is only commented if the Value matches exactly This is used to comment device=... lines in system.ini
@[section] @[section]Key @[section]Key=Value	The ; in front of the corresponding section is removed The ; in front of the Key in section is removed The ; in front of the Key is removed, if the Value matches exactly. This is used to uncomment device=... lines in system.ini
+ [Section]Key= + [Section]Key=Value	Adds Key in Section. Adds Key and Value to section.
&[Section]Key=Value	This Key/Value is always inserted, even if already a key exists, but with a different value. Used to insert device=... lines in System.ini
*[Section]Key=Value	Same as &, but the value is casesensitive Normaly not used.
![Section]Key=Value	A key is inserted in section, but a numeric value is added to Key, representing the highest+1 value of Key. (Only ?) used for Progman.ini to add new program groups. ~ is used to delete such keys if needed. If the value of the command is already found, then the line is not duplicated. You can add multiple ! entrys per Section/Key pair in one pass
>[Section]Key=Value	Adds Value to an existing value. The old Value is preserved and the Value is added added at the end of the line. When the values are separated with a period, (like network= in system.ini) then you must add a period sign before the value. Like >[386enh]networks=,vipx.386. The period sign is then inserted when used.
/[Section]Key=Value	Same as >, but the Value is inserted in front of all existing values. When the values are separated with a period, (like network= in system.ini) then you must add a period sign after the value. Like >[386enh]networks=vipx.386,. The period sign is then inserted when used.
<[Section]Key=Value	Deletes Value from the Key/Value line. Period signs should be handled correctly.
r[Section]Original=NewValue	The string <Original> is replaced with <NewValue>, but only in if [Section] is the current one. If you want to replace the Original in ALL sections (even in the section definition) then specify [] as the section.
R[Section]Original=NewValue	Same as r command, but the case must match.
S[Section]Key=<EnvVarname>	The environment variable with name <EnvVarname> is set to the Value of this Section/Key.

If [Section]Key does not exist, the environment is not modified.
The modification is done in the master environment, which is the one of the FIRST command.com loaded.
If you use this command under windows, the environment of the shell inside windows is NOT modified, but when you leave windows, the environment has the modified values !

Conditional execution

You have multiple IF commands available to execute commandlines based on external values. You can use these commands either in the CMD file, or in the ICM file.

IFMEMBER <GroupName> <Normal commandline>

Is executed if User is member of the give Netware Group. The User must be logged in the server.

IFNMEMBER <GroupName> <Normal Commandline>

As IFMEMBER, but the commandline is only executed when the user is NOT member of the group

IFENV <%EnvVar%> <Value> <Normal commandline>

When the environment variable %EnvVar% is equal to <Value>, the command is executed

IFNEV <%EnvVar%> <Value> <Normal commandline>

Same as IFENV, but NOT EQUAL

IFEXISTENV <EnvVar> <Normal commandline>

When the environment variable EnvVar exists, the command is executed. Please note, that the name of the environment variable is specified without any % signs.

IFNEXISTENV <EnvVar> <Normal commandline>

Same as IFEXISTENV, but NOT EXISTS

DOS Environment Variables

In ALL strings, you can insert a sequence %...%, to replace the text with the content of an DOS environment variable. This is true for the CMD file AND for the ICM file.

```
+ [boot]run=F:\USER\%USER%\WINDOWS\CAL.EXE
```

%USER% will be replaced with the value of the USER - Environment variable.

When It finds a single % in the line then the % remains unmodified.

All %% are replaced with a single %. So you can use F:\USR\%%USER%\TEMP to be transformed into F:\USR\%USER%\TEMP. If you simply use F:\USR\%USER%\TEMP it will try to replace %USER% with the corresponding environment variable.

You must use this, every time you have more than one % sign in the value, but don't want to replace it.

If the specified environment variable does not exist, then a warning message is displayed on the screen, and the %...% is removed from the value.

EXAMPLES

![Section]Key=ValueZ

[Section] Key1=Value1 Key2=Value2 Key3=Value3	[Section] Key1=Value1 Key2=Value2 Key3=Value3 Key4=ValueZ
--	---

>[Section]Key=Value1

[Section] Key=Value0	[Section] Key=Value0 Value1
-------------------------	--------------------------------

>[Section]Key=,Value1

[Section] Key=Value0	[Section] Key=Value0,Value1
-------------------------	--------------------------------

/[Section]Key=Value1,

[Section] Key=Value0	[Section] Key=Value1,Value0
-------------------------	--------------------------------

<[Section]Key=Value1

[Section] Key=Value0,Value1	[Section] Key=Value0
--------------------------------	-------------------------

IFMEMBER MANAGER ![Groups]Group=LAFW.GRP

Adds GroupXX=LAFW.GRP only if the user is member of the group MANAGER

IFENV %USER% SSCA +[Windows]Supervisor=True

Sets Supervisor=True, when the environment variable USER is equal to SSCA

+[boot]run=F:\USER%\%USER%\WINDOWS\CAL.EXE

%USER% will be replaced with the value of the USER - Environment variable.

+[boot]run=F:\USER%\%USER%\%USER%\WINDOWS\CAL.EXE

gives run=F:\USER%\%USER%\WINDOWS\CAL.EXE

r[][OldSectionName]=[NewSectionName]

This command renames a section heading. Maintini then looks in each line for the Old-name and if it finds it, it replaces it with the new name. You should include the [...] to be shure, that only section headings are replaced.

%INICommand%

With this command, the complete command is read from the environment variable INICommand. So when INICommand is equal to +[S]X=Y then this command is executed. You can use this feature in the ICM files too, not only in the CMD files.

Take care when...

- Modifying device= lines in the SYSTEM.INI file. Since this is the only place where multiple keys exists, who have the same name. To handle this special situation, you MUST use the & command instead of the + command. When you try to do modifications with the + command, then either all device= lines have the same value, or/and some keys are added multiple times.

- With the r command it is very easy to move a program/library to a new loaction. Of course you must take care to correct all other references to that location. (In the registration database OLE, in DDE-statements and of course the program icon(s) in the program manager.)

Known Problems

- If your DOS environment is full, then it is possible that maintini crashes the workstation. I haven't found the problem, so I can't do anything against this. So if you expect such problems, then create a greater environment.

- When using the S command to set environment variables, then be aware of the following. The modification is done in the master environment, which is the one of the FIRST command.com loaded. If you use this command under windows, the environment of the shell inside windows is NOT modified, but when you leave windows, the environment has the modified values !!! When I find a TP library which handles this correctly, I will implement it, that the parent environment is changed. (Perhaps you have an idea where to find such a library ???)

- No other Problems are known for this release V1.20

Licence Agreement

MAINTINI is emailware, which means if you find this program usefull, you should send me a email. (Or a nice postcard if you don't have access to a email system)

If you send me a message, then I will inform you of future versions of this utility.

My email address:

```
Compuserve      : 100034,3536 Andre Schild
Internet        : 100034.3536@compuserve.com
AT&T Mail       : mhs!csmail!100034.3536
X400            : /c=US/ad=compuserve/pd=csmail/d.id=100034.3536
MCI             : TO: Andre Schild
                 EMS: COMPUSERVE/ MCI ID:281-6320
                 MBX: 100034,3536
MHS             : MAIL@CSERVE {100034,3536}
```

Postal adress: Andre Schild
Pfeidstrasse 8
CH-2555 Bruegg b. Biel
Switzerland