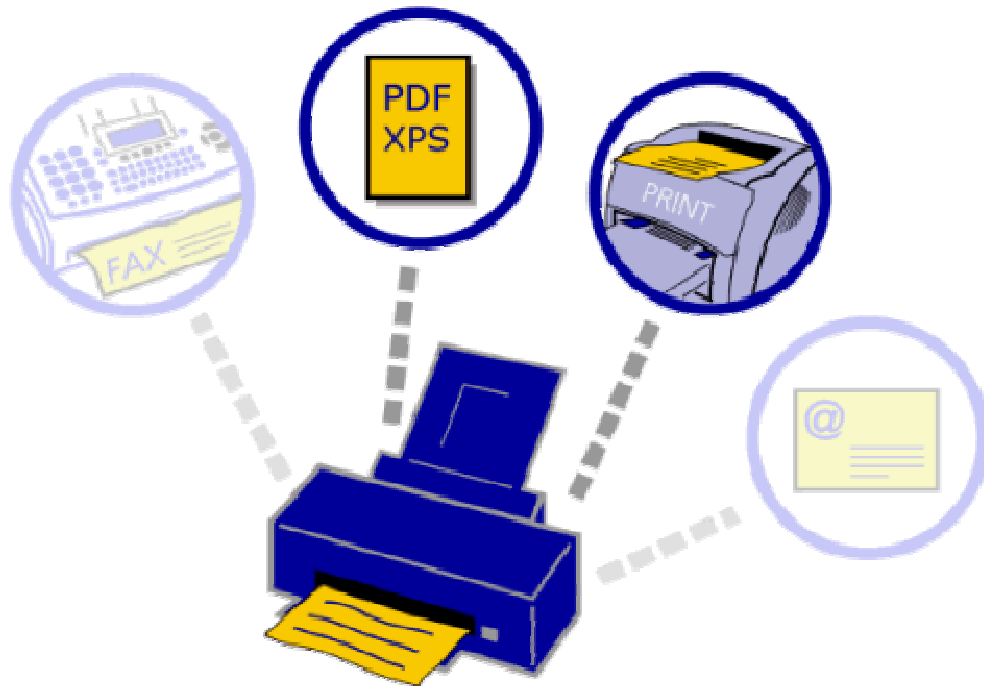


PrintMulti



PrintMultiplier

(Print once – Get multiple output)

Version 1.0.1.3

Dieter Riekert

<http://www.lvbprint.de>

info@lvbprint.de

Content

Introduction.....	3
License Agreement.....	4
Printing with PrintMulti.....	5
Macro expansion	8
Sections and actions in the „printmulti.ini“	10
Common.....	10
Entries for the printers	11
The „Print“ Action.....	12
The „Exec“ Action	14
Network problems	15
Examples.....	16
Logging of print jobs	16
Saving of print jobs	17
PDF-Printing (Ghostscript).....	18
Installation advices	19
Addendum.....	20
Environment variables set, when executing programs	20
Paper formats.....	20
Values for „ExecuteShowWnd“	21

Introduction

PrintMulti is a print processor, which makes it possible to output a print job to multiple printers. For each created jobs varied settings can be carried out.

Additionally programs can be executed on the produced spool files.

Extensive, configurable log files are generated. With this option it is for example possible to detect which user has printed to a specific printer in colour. The format for the log files is csv.

The print output can be influenced in the following ways:

(not all possibilities can be combined)

- Printing in reverse order
- Several pages composed on one sheet (2, 4, 6, 9, 16), optionally with a border around the pages
- Booklet printing with duplex printers
- Simplex, duplex horizontal, duplex vertical
- Changes of the paper format
- Force Black-white / color printing
- Different input bins for the first, following and the last page
- Saving the output into a file or append it to a file
- Execution of programs on the generated files (e.g. for automatic pdf-creation)

PrintMulti is controlled solely with configuration files in ini-fileformat.

***PrintMulti* was not designed for the unexercised user. It is intended for experts with knowledge in printing.**

License Agreement

This is a reduced translation of the original text in German language (from www.lvbprint.de)

Only the original text is legally binding.

Conditions of use:

You may install and use this software ("PrintMulti") without charge on any number of computers running under a Windows Client operating systems (2000, xp, vista) even if it is used as a print server.

To use the software on server operating systems (2000/2003 Server, Terminalserver,...) a license with costs is needed. Installation for test purpose is granted.

The logging functionality is free for all kinds of operating systems, including servers.

To receive server licenses please contact us.

Transfer the software

You may distribute this software with your own software if you regard the conditions of use.

Limited Warranty and Limitation of Remedies:

The program and documentation are provided "as is" and without warranty, express and implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose. In no event will the author be liable for any damages, including lost profits, lost savings, or other incidental or consequential damages, even if the author is advised of the possibility of such damages or for any claim by you or any third party.

Conclusion

The location of the competent court for all legal action in connection with the Software and this contract is D- Karlsruhe if the contract partner is a registered trader or equivalent, or if he has no legal domicile in Germany.

This contract is exclusively governed by the law of the Federal Republic of Germany.

Should any provision of the contract prove unenforceable or if the contract is incomplete, the remaining provisions will remain unaffected. The invalid provision shall be deemed replaced by the provision which in a legally binding matter comes nearest in its meaning and purpose to the unenforceable provision. This shall apply to any omission in the contract that may occur.

Printing with PrintMulti

All relevant data are loaded from a configuration file. The path and name of this file are stored in the registry at key `HKEY_LOCAL_MACHINE\SOFTWARE\VBPrint\PrintMulti`, Entry `ConfigurationFile`. If no entry is found, a file "printmulti.ini" is searched in the windows-directory.

The installation copies an example file to the path „<program files>\Printmulti“ and sets the entries in the registry.

To use the possibilities of PrintMulti, it is recommended to install a local printer, which prints to a single file (see the example)

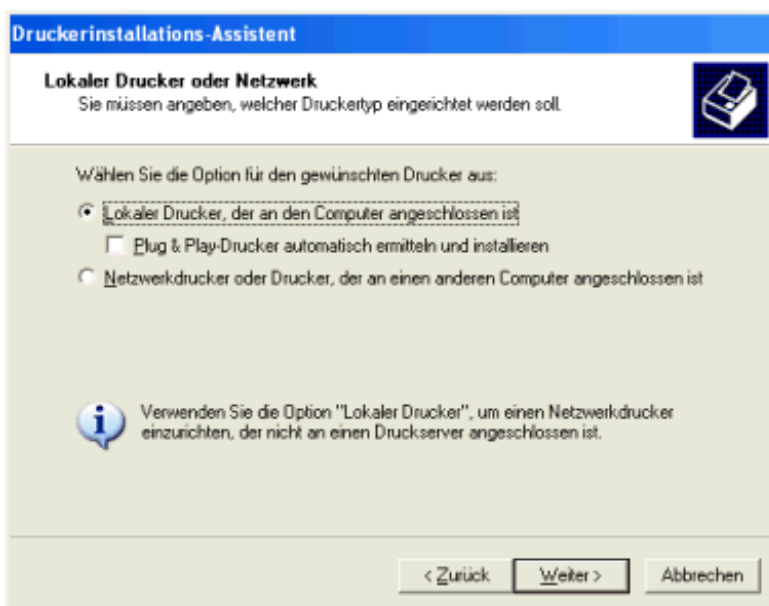
The following conditions must meet, so that PrintMulti comes in action:

1. The PrintProcessor „PrintMulti“ must be assign for the printer in the advanced options dialog.
2. The relevant configuration file must contain a section for the printer, the Entry `Active` must be set to "1" and at least one action has to be defined.

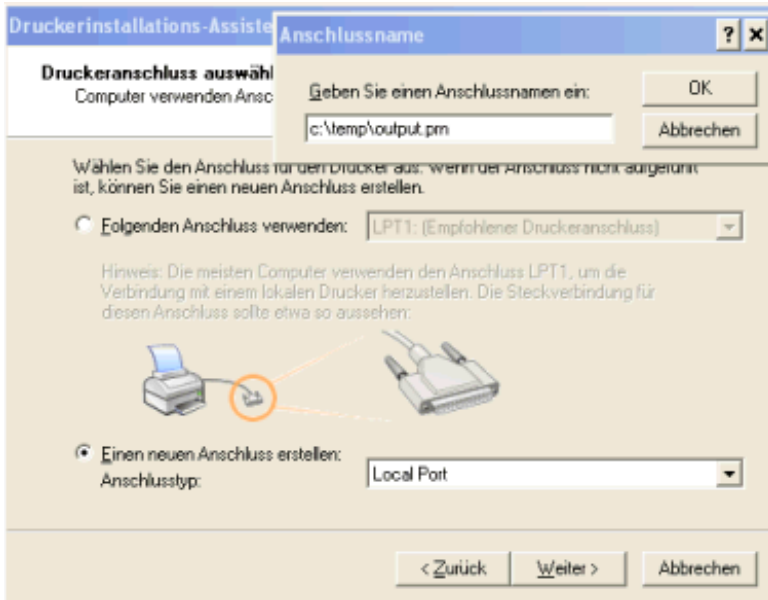
If you have the problem, that nothings happens when printing, then check these conditions first

The following pictures show how to install a printer "*MiniPhotos*", which causes a print job to be send to the printer "*PhotoPrinter*" twice. The first time the original output and the second time with nine pages compressed on one sheet in black/white.

Some intermediate steps are ignored.

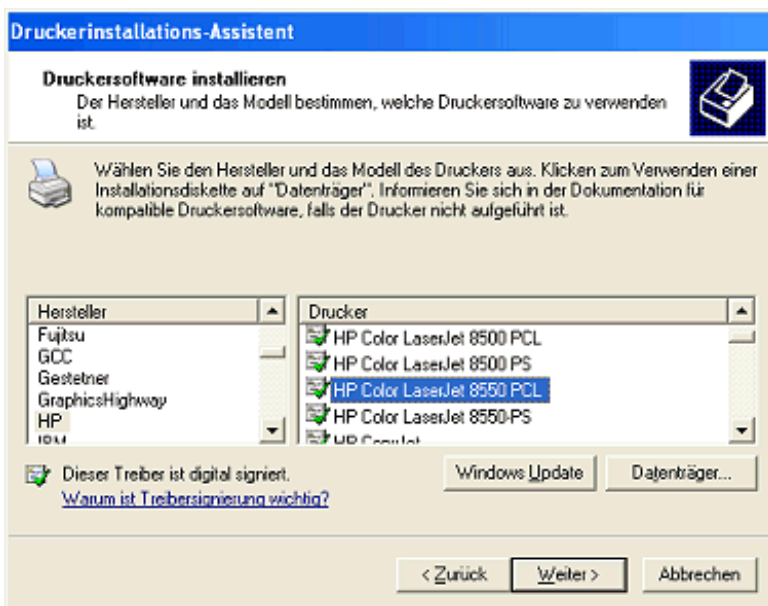


Install a new printer



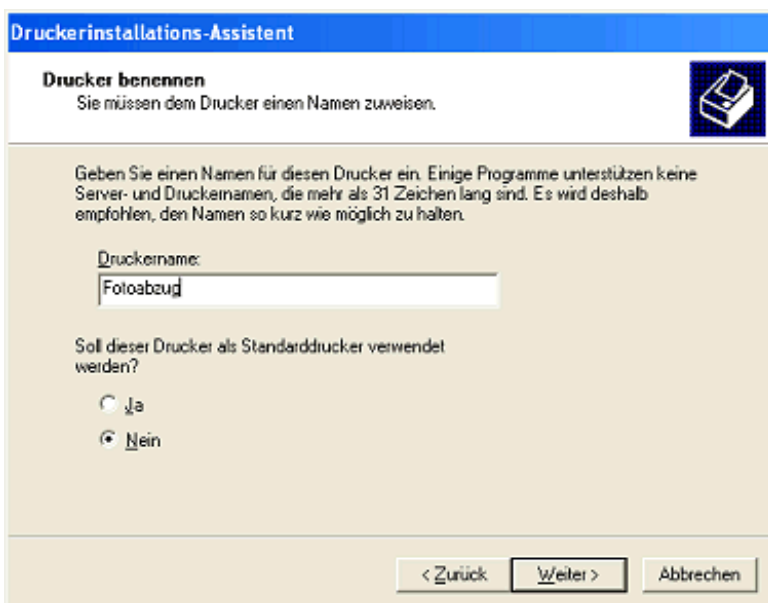
As port use e.g. a new “Local Port” with a fix file name as destination (If you share the printer all users must have write access to this file!). All output to the printer is saved to this file overwriting the content each time.

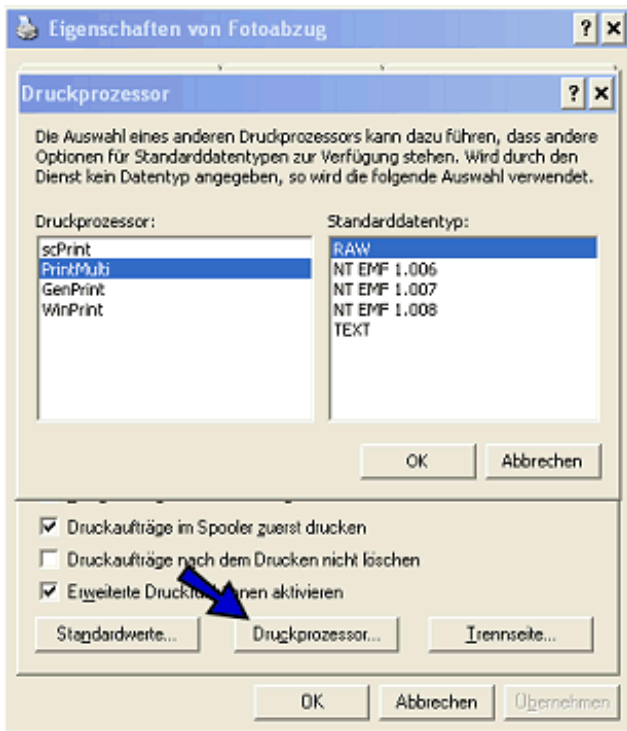
After the printer is configured with „PrintMulti“, there will be no print output to this printer anymore.



Choose a printer which supports many paper format, color printing and duplex.

There is no experience which printer drivers behave best. It should be independent. But if you meet a problem, try to use the same driver as the physical printer.





Do not forget to enable the duplexing feature if you want to use duplex mode.

PrintMulti can now be set as print processor. The data type can stay "Raw" as suggested.

[Common]

LogJobMask=4
LogJobFile=d:\temp\jobs.csv

LogMask=27
LogFile=d:\temp\debug.csv

[MiniPhotos]

Active=1
Action1=Print;ActionPhotoPrinter1Co
Action2=Print;ActionPhotoPrinter9BW

[ActionPhotoPrinter1 Co]

Active=1
Printer=PhotoPrinter
nUp=1

[ActionPhotoPrinter9BW]

Active=1
Printer=PhotoPrinter
Color=0
nUp=9
nUpBorder=1

Minimal configuration file

„C:\Program Files\PrintMulti\printmulti.ini“

Two log files are written – one with the job data and one with extensive debugging information.

Each job to the printer „MiniPhotos“ is send to the two actions in the given order, if Active is set to 1 (there will be no output to the printer MiniPhotos even, if it is attached to a physical port).

If Active=0 (in MiniPhotos section) the printer will behave as if no PrintMulti is installed. But Logfiles will be created anyway..

Macro expansion

„#T;#(06)J;#(-20)P;#(07)C;#(-20)U;#(-20)M;#(-30)D;#(5)Z;#(5)c;#(-6)B;#(-10)A;#(-20)E“

This string will be used for each debug entry, if not defined otherwise.

The following rules apply for string substitution:

- A macro starts with ‚#‘ optional followed by a format expression in brackets and terminated by a letter e.g. „#(...)Z“ or
- An environment variable surrounded by the ‚%‘ character e.g. „%TEMP%“. Only system environment variable can be used (no user environment variable).

The table shows the different macro letters.

Macro	Type	Meaning
Sonstige und Informationen aus den Jobdaten des Druckjobs		
A	String	Name of the action (e.g. „ActionPhotoPrinter1Co“ in the example above)
B	String	The type of the action („print“ or „exec“ at the moment)
C	DWORD	Printerspecific counter (for the Printmulti aware printer). This value will be increased with each print job. It is stored in the registry.
D	String	The name of the document, which is delivered by the application.
K	String	The name of the document with invalid characters for path names replaced by an underline. Invalid characters are: (<>:“^)
E	String	Action specific text, which will be replaced with the name of the destination printer in a „print“-action. (e.g: „PhotoPrinter“).
F	String	File name of the internally copied spool file
G	String	Action specific file name. In a „print“-action, when output goes to a file, this file name will be used. Otherwise the name of the spool file.
g	String	Like „G“, without file extension (e.g. „c:\temp\file“)
J	DWORD	Job id of the print job (not unique, will be reused by windows)
M	String	Name of the computer which starts the print job
P	String	Name of the main (PrintMulti) printer (e.g. „MiniPhotos“)
S	Date/Time	Time, when the job was created
T	Date/Time	Current time
U	String	Name of the user who printed
Z	DWORD	Number of pages, extracted from the print job data (not calculated)
z	DWORD	Total number of copies (only for action „Print“)
Data from the devicemode of the print job		
c	0/1	colour, 0 = black/white, 1 = colour (dmColor – 1)
b	DWORD	Paper source (dmDefaultSource)
d	0/1/2/3	0 = not set, 1 = simplex, 2 = duplex vertical, 3 = duplex horizontal (dmDuplex)
o	0/1/2	0=nicht gesetzt, 1 = portrait, 2 = landscape (dmOrientation)
s	DWORD	Paper size (z.B. 8 = A3, 9 = A4) (dmPaperSize), 0 = not set
y	DWORD	resolution (dmYResolution), 0 = not set

The permitted format expressions are dependent from the type in the table above. The following formats are configurable for the different types. To understand them, you have to know, how the printf format works inside the C-programming language.

String:

Example: myTest“

Standard: existing string, e.g. „#D“ → “myTest”

Format: Embedded in printf with „%<format>s“, e.g. „#(-10)D“ is replaced by „%-10s“, in the example „myTest “. The characters „%“ and „#“ are forbidden in the format string and lead to a standard format.

DWORD or number:

Beispiel: 40000

Standard: „%u“ e.g.. „#C“ → „40000“

Format: If the last character in the format string is one of „diuxXo“, then it will be used for formatting. Otherwise , „%u“ is used.
“#(08X)C“ → “%08X“ the example will expand to „00009C40“,
“#(08)C“ → „%08u“ expands to „00040000“. „%“ in the string is forbidden and lead to a standard format.

Date / Time:

Example: 1964-11-23 13:45:12

Standard: from „printmulti.ini“ or “%Y-%m-%d %H:%M:%S”, if not configured

Format: will be formatted with the function „strftime“. Please search the internet for a description.

Sections and actions in the „printmulti.ini“

The configuration file will be reread with every print job. So changes will affect immediately with the next print job.

Common

The common-section is mostly used to set options for the logging output. It is possible to handle to independent log files. The file names can contain macros from the last section with the exception of devicemode macros.

e.g. „C:\LogFiles\#P\#U_#(%Y_%m)S.csv“ creates for each printer and each user every month a new file. In the example before: “C:\LogFiles\MiniPhotos\dieter_2007_05.csv”.

Most of the possible values are strings, which can contain macros. All characters, which do not belong to a macro, will be used unmodified. Additionally it is possible to defined the date/time-format, which will be used with the macros „#T“ and „#S“ if no format string is given.

For the three output masks the bitfield has the following meaning:

0	No output
1	Error
2	Warning
4	Jobinfo
8	Info
16	More info

Example:

```
[Common]
;Bitmasks:1=Error,2=Warning,4=Jobs;8=Debug;16=More Debug
;Used with DbgView from sysinternals
DbgOutMask=255

;two seperate debugfiles, each with a different bitmask. No file->no output
LogJobMask=4
LogJobFile=d:\temp\jobs.csv
LogJobFileFormat=#T;#(06)J;#(-20)P;#(07)C;#(-20)U;#(-20)M;#(-30)D;#(5)Z;#(5)c;#(-6)B;#(-
• 10)A;#(-20)E
LogJobHeader=Type      ;Date/Time      ;JobId ;Printer      ;Counter;User
• ;Machine      ;Documenttitle
• ;Pages;Color;Action;ActionName;Actionmessage      ;JobMessage
LogJobHeaderOut=y

LogMask=27
LogFile=d:\temp\#P\debug.csv
LogFileFormat=#T;#(06)J;#(-20)P;#(-5)B;#(-10)A
LogFileHeader=Type      ;Date/Time      ;JobId ;Printer
• ;Action;ActionName;Message
LogFileHeaderOut=y
LogStdDateFormat=%Y-%m-%d %H:%M:%S
```

The lines beginning with „• “ are the continuation of the last line and are only wrapped here.

(The LogJob... and LogFile... entries only have different names. They behave the same. The table lists only one type. The standard entries are shown in the example above.

Folgende Einstellungen sind verfügbar:

Schlüssel	Standard	Bedeutung
DbgOutMask	0	Bitmask for output messages to an attached debugger
LogJobMask	0	Bitmask for output messages to a file
LogJobFile	„	Output file. If no valid filename is defined (can contain macros) nothing will be written !!
LogJobFileFormat	See example	Format for debug messages
LogJobHeader	See example	A header, which will be written as first line in new files, if the next value is “true”
LogJobHeaderOut	True	Output the header or ignore the „LogJobHeader“ entry
LogStdDateFormat	Siehe Beispiel	Date/time format used, if no other format is given in time relevant macros

The debug message itself will be appended to the text defined with “LogJobFileFormat” or “LogFileFormat”.

If the debugmask contains the bitmask value „JobInfo”, then for each print job a line is added with the status “FAILED” or “OK”, followed by the action name “Print”, “Exec”, ... and the duration of the job in ms.

For the main print job a debug entry is added after all action entries. It contains “OK”, if all actions were successfully. Action type and name are empty for this entry.

Entries for the printers

As mentioned earlier, if you want to use the „PrintMulti“-features, you have to use „PrintMulti“ as print processor, create a section in the configuration file with the name of the printer and set the entry „Active“ to *yes* (can also be 1 or start with a ‘y’ or ‘j’ character). The default for “Active” is false.

If the value is false (=0, or the first character ‘n’), then the original function (winprint) will be called. The log entries will be still created. So it is possible to use printmulti solely as log output medium, even for Windows 2003 Server without a licence.

The printer section may contain action entries, which are executed in the order, they are listened in the file. An action entry has to start with the word “Action” (not case sensitive). The rest is arbitrary. On the right side of the equal sign you define the type and the section of the action. The action will only be executed, if the entry “Active” is set to true in the action section. That is also the default for this entry (opposite to the default in the printer section).

The „Print“ Action

The following adjustments affect the printing to the „slave“-printers:

Aktion Print			
Einstellung	Type	Default	Bedeutung
Active	Bool	1	Use the action ?
Printer	String	Name of the section	Printer name to print to
UseSystemAccount	Bool	0	Print with the system account (normally not necessary).
nUp	Int	*0)	Number of pages per sheet. (1,2,4,6,9,16)
nUpBorder	Bool	*0)	Border around the pages
Booklet	Bool	*0)	Do booklet printing
Duplex	String	Simplex	„s“ or „S“ für Simplex, „v“ or „V“ for Duplex vertikal and „h“ or „H“ for Duplex horizontal
Reverse	Bool	*0)	Print last page first. Does not work with all other options.
Collate	Bool	*0)	Change page order: 1-2-3-1-2-3 in place of 1-1-2-2-3-3)
Color	Bool	From Devmode	Force black/white printing for color printers
DrvCopies	Int	*1)	Number of copies to set in the devicemode
TotalCopies	Int	*1)	Total number of copies
PaperSize PaperSizeConversion PaperSizeConvertAlways	Int String Bool	*2)	Paper size
PaperSource1 PaperSourceN PaperSourceL PaperSourceConversion	Int Int Int String	*3)	Paper source for different pages
Save2File	String	„“	Print output will be saved to the file. The file name can contain macros.
Append2File	Bool	False	Append to the file from (for Save2File).
Execute ExecuteCmd ExecuteCurDir ExecuteAddPath ExecuteFlags ExecuteTimeout ExecuteAsUser ExecuteShowWnd	Bool String String String Int Int Bool Int	*4)	Options for the execution of programs on the created RAW-files

- *0) The standard for these values is extracted from the job attributes while printing. Some printer drivers offer the possibility to set the number of pages per sheet and the duplex printing feature (only for duplex printers) in the printer properties dialog.
- *1) The number of copies should not be copied from the main print job. For example if you want to archive the print output, then it is not desirable to save multiple copies to the archive, if the user had chosen to print multiple copies.
The two values „*DrvCopies*“ and „*TotalCopies*“ controls the copy behaviour. If not set, both values contain the number of copies defined when printing.
„*TotalCopies*“ is total number of copies that will be printed to the slave printer. „*DrvCopies*“ the value set inside the devicemode. The printer driver decides what to do with this number. PrintMulti reprints the job until the number of copies are reached.
Example: TotalCopies 7, DrvCopies 3, result in 3 print passes with 3, 3 and 1 copy in the devicemode.
- *2) Often the destination printers do not provide the same paper sizes than the main printer (e.g A3 with A4 printers).
If you set „*PaperSize*“, than this value will be used for all pages for the destination printer. If the value is invalid than the standard value from the printer is used.
With „*PaperSizeConversion*“ it is possible to define a map between source and destination paper sizes. The format consist of the source size followed by „->“ and the destination size. Multiple map entries are separated by a semicolon. A source size of „0“ defines the standard size, which is used when no mapping is defined.
Example: „A3->Exe;0->A4;1->Letter;A4->B5“. The text representation or the accordingly constants (see the addendum) can be used to define the paper sizes.
If „*PaperSizeConvertAlways*“ is true, than the conversion will always be used. In the other case only if the source format does not exist in the destination printer.
- *3) The selection of the paper bins applies similar to the paper formats. If values for the first, the following and/or last page are set, than these are used.
The values and papier bin names are unfortunately dependent from the printer driver. This complicates the mapping. Names are always searched in the context of the destination printer. First the exact string is scanned and if not found the beginning of the name is tested for identity.
Example: „manu->lower;3->2;15->auto“ (replaces e.g.: „manual“ oder „automatic“)
- *4) If „*Execute*“ is set to true, than a process is called after finishing the print job. If the output was saved into a file („*Save2File*“), than the macro „#G“ bzw. „#g“ will assigned to this file name, otherwise the macros are replaced by the name of the spool file.
„*ExecuteCmd*“ contains the commandline for the executing process. The current directory can be adjusted with the command „*ExecuteCurDir*“.
„*ExecuteTimeout*“ describes the time in ms to wait for the process to terminate. A value of „*INF*“ (= INFINITE) waits forever, whereas „0“ returns immediate.
For the values of „*ExecuteFlags*“ and „*ExecuteShowWnd*“ the windows api reference “CreateProcess” should be consulted. A useful value for „*ExecuteFlags*“ could be 0x08000000, which leads to suppression of the console window when executing batch-files.
If „*ExecuteAsUser*“ is false, than the process is executed with the „*System*“ account. Otherwise in the context of the printing user. „*ExecuteShowWnd*“ will normally be set to 1 (SW_SHOWNORMAL).
Attention!!! The process can't access the registry key „*HKEY_CURRENT_USER*“. Entries are read from „*HKEY_USERS\DEFAULT*“. Perhaps it is necessary to copy some value to this destination (e.g. Database access).
In the environment for the calling process some variables are set. Please consult the addendum for these values. Additionally the value from the variable „*ExecuteAddPath*“ is appended to the „*PATH*“-environment variable.
This is all useful to create PDF-files with Ghostscript (look at the examples).

The „Exec“ Action

With this action it is possible to run a program on the spool file itself. With the help of the SpIViewer a preview for nearly all applications can be realized. With the viewer the print job can be print to arbitrary printers with some options.

If the action is executed in a separated thread to free the printer for other jobs as soon as possible, the spool file will be copied.

PrintMulti waits in any case until the process has finished. This is necessary to manager the spool files properly. If no separated thread is used, the printer is blocked until the calling application has finished (e.g. the viewer closed).

Here are the options in detail:

Aktion Exec			
Einstellung	Type	Default	Bedeutung
Active	Bool	1	Use the action ?
NewThread	Bool	False	Execution in a seperate thread
ExecuteCmd ExecuteFlags ExecuteAsUser ExecuteShowWnd	String Int Bool Int	See action „print“s	Settings for execution

Please notice, that the executed programs normally do not have access to network resources or to the registry path HKEY_CURRENT_USER.

If you copy the settings from

HKEY_CURRENT_USER\Printers

to

KEY_USER\.Default\Printer

the problem can be avoided.

Example:

```
[ActionPreview]
Active=1
ExecuteCmd=%ProgramFiles%\SplView\splview "#G"
;Flags for CreateProcess, 0x08000000 e.g. create no window for console app
ExecuteFlags=0
NewThread=y
;run as the printer user
ExecuteAsUser=y
;ShowCmd for ShowWindow (HIDE=0,Minimize=2,Maximize=3,...)
ExecuteShowWnd=3
```

Network problems

A print processor is a DLL, which is loaded by the spooler process „spoolsv.exe“. The spooler runs with the system account.

The systemaccount has many rights, but also some restrictions:

1. Uses the standard user profile (HKEY_USERS\DEFAULT). It could be necessary to copy settings from the user registry path .
2. Access to network resources is only possible, if the allow null sessions. Look here to learn about that: <http://support.microsoft.com/kb/289655>

No access to network printers:

If the access does not work, you have three possibilities.

1. Install the printer locally. As new „Local Port“ use the printer share on the server with UNC-names.
2. Copy data from
“HKEY_CURRENT_USER\Printers\Connections“
“HKEY_CURRENT_USER\Printers\DevModePerUser“
“HKEY_CURRENT_USER\Printers\DevModes2“
to
“HKEY_USERS\DEFAULT“
in the appropriate path. This does not work always!
3. Experiment with the settings „UseSystemAccount“ in the printer section and try to allow the null session access.

Examples

Logging of print jobs

Let us assume, you want to log in a database for each print job, which user has printed it on which printer and how many pages in color or blackwhite the job had. Surely, you can examine the log files, but in this example we use a program executing to do that.

Install a printer with name „SumPages“ and assign „PrintMulti“ as print processor.

The Ini-Section looks like this:

```
[SumPages]
Active=1
Action=Print;SumPages
Execute=yes
ExecuteFlags=0x08000000
ExecuteCmd=%windir%\sumpages.bat "#U" "#P" #c #Z #z
```

“sumpages.bat” would contain code to insert the parameters into a database table. Here I use a simple batchfile, which appends the parameters to a CSV-file.

```
echo %1,%2,%3,%4,%5 >>%windir%\pages.csv
```

The destination file contains for every printjob one line with the user, the printer, 1 for color / 0 for bw, the number of pages and the number of copies.

Some further comments:

- The action points to itself – acts as a main entry and action entry. A direct recursive call of PrintMulti will be suppressed internally. To use this construct is a little dangerous und could lead to an endless loop. So be carefully or better **avoid** it.
- Since no “printer”-entry was set, the name of the section “SumPages” will be used as printer name.
- After the printing the command will be execute. The flags cause no console window to pop up.

Saving of print jobs

You want for example save all print jobs, so that they can later be reprinted or send to dirty party companies to print them.

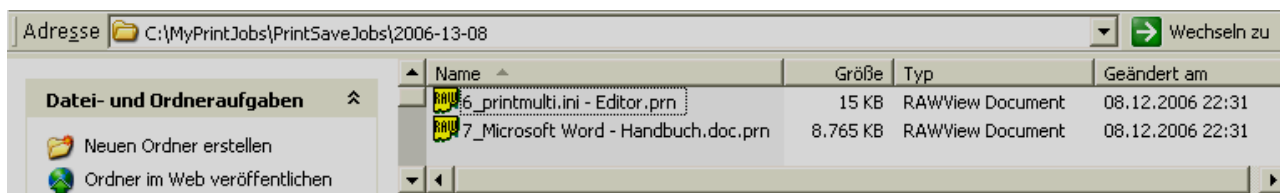
```
[PrintSaveJobs]
Active=1
Action1=Print;PrintJobs
Action2=Print;SaveJobs

;print the jobs
[PrintJobs]
Active=1
Printer=RealPrinter

;save the jobs and create a logfile
[SaveJobs]
Active=1
Printer=RealPrinter

;Force one copy
TotalCopies=1
DrvCopies=1
Save2File=c:\MyPrintJobs\#P\#(%Y-%m-%d)S\#C_#K.prn
Append2File=0
Execute=yes
ExecuteFlags=0x08000000
ExecuteCmd=%windir%\savepages.bat
```

The files would be saved like this:



The batchfile „savepages.bat“ contains only one line:

```
echo "%PM_PRINTTIME%" ; "%PM_USER%" ; "%PM_PRINTER%" ; "%PM_FILE%" >>c:\MyPrintJobs\jobs.csv
```

The job file would contain after the two print jobs:

```
"2006-12-08 22:31:29";"dieter";"PrintSaveJobs";"c:\MyPrintJobs\PrintSaveJobs\2006-13-08\6_printmulti.ini - Editor.prn"
"2006-12-08 22:31:52";"dieter";"PrintSaveJobs";"c:\MyPrintJobs\PrintSaveJobs\2006-13-08\7_Microsoft Word - Handbuch.doc.prn"
```

The example shows the usage of the environment variable, which are listed in the addendum.

By the way, „#D“ in the file name could give some problems. If you print for example a document from an editor, the document name could contain the full path of the printed file. If you use this in a “Save2File”-line it will lead to an invalid filename. Please use “#K” instead of “#D” in this case.

Now every day a new directory will be created and all jobs, that are printed to the printer „PrintSaveJobs“ were stored there. The CSV-File contains references and information about the file and could be analyzed.

If you set „Append2File“ to 1, than the data would be appended to the file. This could be useful if you give the data to another person to print more jobs at once.

“#C” is a unique job counter for the printer “PrintSaveJobs” in the example, which is increased with every print job. It is stored in

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Printers\PrintSaveJobs\PrinterDriverData\JobCounter"
and can be reset manually here.

PDF-Printing (Ghostscript)

To create PDF documents with Ghostscript an indirection over a postscript driver is necessary. The print output of the postscript-printer is converted to PDF by calling the batch file „ps2pdf“. Help for that can be found in the file “Ps2pdf.htm” in the subdirectory “doc” of the Ghostscript-installation.

The inf-file to install a suitable postscript driver can be found in the subdirectory “lib” of the Ghostscript-installation. It normally installs the printer “Ghostscript PDF”.

Because many tools use this way to create PDF files, there is a lot of information available in the internet (e.g. “redmon” is often used as tool. Most things that apply to redmon, are also valid for PrintMulti.)

```
[PDFPrinterFine]
Active=1
Action1=Print;ActionCreatePDFFine

[ActionCreatePDFFine]
Active=1
Printer=Ghostscript PDF
;will not be deleted, so use always the same name
Save2File=%TEMP%\out.ps
Execute=yes
ExecuteFlags=0x08000000
;need path to find executables
ExecuteAddPath=%ProgramFiles%\gs\gs8.54\lib;%ProgramFiles%\gs\gs8.54\bin
ExecuteCmd=%ProgramFiles%\gs\gs8.54\lib\ps2pdf14.bat "-dPDFSETTINGS=/prepress" "#G"
"c:\PDFOutput\#C_#U_fine.pdf"
```

Creates PDF-files with high resolution in the subdirectory „c:\PDFOutput“. If the “lib” and “bin”-path of the Ghostscript-installation are not contained in the environment path, they must be set by the “ExecuteAddPath” entry.

If you meet a problem, than insert before the ghostscript call the instruction “cmd /k”. After printing a console window should be opened, which shows the Ghostscript commands („ExecuteCmd=cmd /k %programFiles%\gs...“). „ExecuteFlags“ must be set to 0 for the window to open.

If further processing the PDF file is desired, you have to create a script file, which contains to PDF creating and the later processing.

Installation advices

The installation with the setup program of PrintMulti saves an example configuration file „printmulti.ini“ in the directory „%ProgramFiles%\PrintMulti“.

To use the examples some printers has to be installed. You should choose printers, that give a lot of possibilities (Color, DIN A3, Duplex).

The printers “TestPrinter”, “PDFPrinterFine” and “TestAll” should be installed. Dependent on the active entries in the “TestAll” section, there should be a PDF-file in then System-Temp-path created, the output saved to a postscript file, a output to “Testprinter” and a preview shown, after printing something to the “TestAll” printer.

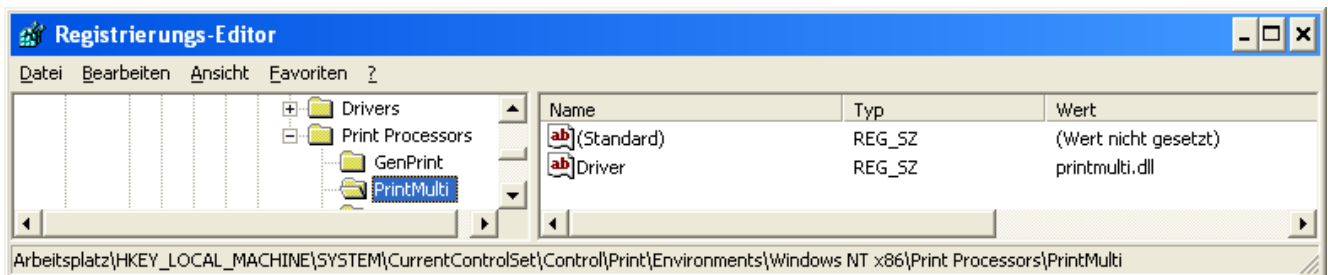
Further you need Ghostscript (in the standard path) and the Ghostscript printer driver. If you meet problems look at the log-files (pathes in the common-section of the configuration file).

„PrintMulti“ can easily installed with the delivered installation program and uninstalled through “System/Software”. The update and deinstallation of a print processor is a little bit problematic. Here are some hints:

„PrintMulti“ consist of the DLL „printmulti.dll“ and the described configuration file. The rest of the files in the installation packet are documentation and help files.

If you want to integrate the installation in your software, it is sufficient to copy the DLL into the „%systemroot%\system32\spool\prtprocs\w32x86“ –path and call the Api-function “„AddPrintProcessor()“. It creates the following entry in the registry:

Falls Sie die Installation in Ihre Software integrieren wollen, genügt es die DLL ins Verzeichnis



The entries could be manually added or deleted, too. But in this case the spooler-service has to be restarted or the computer rebooted.

For a update the spooler should be stopped, the „printmulti.dll“ changed and the service started again.

For a deinstallation all printers, which has “PrintMulti” as printprocessor must assigned a different one e.g. “WinPrint”. Additionally the print processor inside the spool jobs has to be replaced. This is only possible with programming. A call to the Api-function “DeletePrintProcessor()”, which would be the right way, would fail otherwise. Than you should consider to delete the registry entries.

Addendum

Environment variables set, when executing programs

PM_ACTION	Name of the action (#A)
PM_COUNTER	Unique counter (#C)
PM_JOBID	JobId from Windows (#J)
PM_USER	Username (#U)
PM_MACHINE	Computername of the printing computer (#M)
PM_PRINTER	Name of the main printer (#P)
PM_PAGES	Number of pages (#Z)
PM_COLOR	Color printing ? (#c)
PM_PAPERSIZE	Papersize (#s)
PM_DUPLEX	Simplex, Duplex Horizontal or Simplex Vertikal (#d)
PM_ACTTIME	Actual time (#T)
PM_PRINTTIME	Time from the print job (#S)
PM_FILE	Output file (#G)

Paper formats

The following paper format are recognized. The text is replaced with the number in the first column

Wert	Papierformattexte
1	Letter
2	Letter Small
3	Tabloid
4	Ledger
5	Legal
6	Statement
7	Executive
8	A3
9	A4
11	A5
12	B4 (JIS)
13	B5 (JIS)
14	Folio
15	Quarto
16	10x14
17	11x17
18	Note
19	Envelope #9
20	Envelope #10
21	Envelope #11
22	Envelope #12
23	Envelope #14
24	C size sheet
25	D size sheet
26	E size sheet
27	Envelope DL
28	Envelope C5
29	Envelope C3
30	Envelope C4
31	Envelope C6
32	Envelope C65
33	Envelope B4
34	Envelope B5
35	Envelope B6
37	Envelope Monarch
27	DL
28	C5
29	C3
30	C4
31	C6
32	C65
33	B4
34	B5
35	B6
37	Monarch

Values for „ExecuteShowWnd“

You have to use the numbers

SW_HIDE	0
SW_SHOWNORMAL	1
SW_NORMAL	1
SW_SHOWMINIMIZED	2
SW_SHOWMAXIMIZED	3
SW_MAXIMIZE	3
SW_SHOWNOACTIVATE	4
SW_SHOW	5
SW_MINIMIZE	6
SW_SHOWMINNOACTIVE	7
SW_SHOWNA	8
SW_RESTORE	9
SW_SHOWDEFAULT	10
SW_FORCEMINIMIZE	11