

PrintMulti

(Einmal Drucken - mehrfache Ausgabe)

Version 1.0.4.0 -

Dieter Riekert

<http://www.lvbprint.de>

info@lvbprint.de

Inhaltsverzeichnis

| | |
|--|----|
| Einführung | 4 |
| Lizenzbedingungen..... | 6 |
| Änderungen | 7 |
| Version 1.0.4.0..... | 7 |
| Version 1.0.3.3..... | 7 |
| Version 1.0.3.2..... | 7 |
| Version 1.0.3.0..... | 7 |
| Version 1.0.2.1..... | 7 |
| Technische Hintergründe, Einschränkungen | 8 |
| Technische Hintergründe..... | 8 |
| Der EMF-Druckvorgang..... | 8 |
| Der RAW-Druckvorgang | 9 |
| PrintMulti-Drucken | 10 |
| Einschränkungen und bekannte Probleme..... | 11 |
| Installation | 12 |
| Drucken mit PrintMulti | 14 |
| Freigeben von PrintMulti-Druckern | 17 |
| Expandierung von Makros und bedingten Ausdrücken | 18 |
| Makros | 18 |
| Berechnete Ausdrücke..... | 21 |
| Allgemeine Regeln | 21 |
| Tabelle von Operatoren | 22 |
| Beispielumwandlung eines Ausdrucks in einen stackbasierten Ausdruck..... | 24 |
| Bereiche und Aktionen in der „printmulti.ini“ | 25 |
| Common | 25 |
| Einträge für den PrintMulti-Drucker | 26 |
| Die Aktion „Print“ | 28 |
| PaperSize-Einstellungen..... | 29 |
| PaperSource-Einstellungen..... | 30 |
| Execute-Einstellungen | 30 |
| Neu: Textextraktion..... | 30 |
| Verwendung gespeicherter Druckereinstellungen (Schalter: Devmodel) | 30 |
| Die Aktion „Exec“ | 34 |
| Die Aktion „SaveEmf“ | 35 |
| PDF- und XPS Erzeugung mit PrintMulti | 36 |
| Verwendung von zweistufen PDF-Druckern | 36 |
| Verwendung von „nativen“ PDF/XPS -Druckern | 37 |
| Direkte Erzeugung mit PrintMulti (Ghostscript)..... | 37 |
| Netzwerkprobleme..... | 39 |
| Beispiele | 40 |
| Beispiele zu bedingten Ausdrücken | 40 |
| Seitenzahlabhängiges Drucken..... | 40 |
| Benutzerabhängiges Drucken | 40 |
| Dokumentabhängiges Drucken..... | 41 |
| Datumsabhängiges Drucken | 43 |
| Nutzen des Seitentextes | 44 |
| Spezielles Protokollieren von Jobs | 45 |
| Speichern von Druckjobs..... | 46 |
| Verarbeitung von PDF Dateien | 47 |
| Anhang..... | 50 |
| Umgebungsvariable bei der Ausführung von Programmen | 50 |
| Papierformate..... | 50 |
| Werte für „ExecuteShowWnd“ | 51 |

Einführung

PrintMulti ist ein PrintProcessor, der es ermöglicht einen Druckauftrag auf mehreren Druckern auszugeben. Es kann sich dabei um Drucker verschiedenen Herstellern und auch besondere Drucker wie PDF-Erzeuger oder Faxdrucker handeln. Für jeden der weitergegebenen Druckaufträge können vielfältige Einstellungen vorgenommen und Scripte auf den erzeugten Daten ausgeführt werden. Ein automatisches Speichern des Druckauftrags in einem der Grafikformate Tiff (auch mehrseitige), Gif, JPG, Bmp und Png ist ebenfalls möglich.

Dadurch sind zum Beispiel folgende Problemlösungen möglich:

Durch das mehrfache Ausdrucken auf demselben Drucker mit Papier aus verschiedenen Eingabeschächten ist eine automatische Kopiererstellung möglich.

PDF Erstellung über zusätzliche PDF-Drucker (z.B. Amyuni) oder mit Hilfe des mitgelieferten vbs-Scriptes, das ein installiertes Ghostscript nutzt und Unterstützung für PDF/A bietet. Das Script könnte erweitert werden um anschließend ein Archivsystem anzusteuern (Volltextextraktion durch Freewareprodukte aus der erstellten PDF-Datei).

Abspeichern und Weiterverarbeitung der Druckausgaben als Multipage-Tiff Format.

Sie können auch die konfigurierbaren Logdateien (CSV-Format) auswerten und daraus Rückschlüsse über die Art und Anzahl von Druckjobs ziehen (Anzahl Seiten, Farbe/SW, Drucker, Benutzer,...)

Sammeln und Speichern von Druckausgaben im Raw-Format zur Weitergabe an Druckdienstleister.

Umgehen der rudimentären Möglichkeiten zum Speichern oder Anhängen von Druckausgaben in feste Dateien (Windows bietet den FILE-Port - dann muss ein Dateiname ausgewählt werden oder den Druck in eine feste Datei, die allerdings mit jedem erneuten Druck überschrieben wird).

Alle Einstellungen können Bedingungen enthalten. So kann z.B. ein Ausgabedruker abhängig von der Anzahl Seiten, dem Papierformat, dem Benutzer, dem Dokumentennamen, ... ausgewählt werden.

Mit dem zusätzlichen Tool „devmode2file“ können Sie druckerspezifische Einstellungen speichern und mit *PrintMulti* verwenden. Dadurch ist es z.B. möglich, Ausgabeschächte zu definieren oder Wasserzeichen zu verwenden, sofern der Druckertreiber Ihres Druckers das anbietet.

Sie können den ausgedruckten Text von Dokumenten zur Beeinflussung von Konfigurationseinstellungen oder innerhalb von Scripten benutzen.

Hier eine Liste der Möglichkeiten zur Beeinflussung der Druckausgabe auf den zusätzlichen Druckern (nicht alle Möglichkeiten sind kombinierbar):

Letzte Seite zuerst drucken

Mehrere Seiten auf ein Druckblatt (2, 4, 6, 9, 16) ggf. mit Rahmen um die einzelnen Seiten

Booklet Druck

Simplex, Duplex horizontal, Duplex vertikal

Papierformatänderungen

Schwarzweiß/Farbdruck

Unterschiedliche Schachtauswahl für erste, folgende und die letzte Seite

Speichern der Druckausgabe in eine Datei oder Anhängen an eine Datei

Zusätzlich Ausführen von Programmen auf die erzeugte Datei (z.B. PDF-Erzeugung)

***PrintMulti* wird momentan rein über Konfigurationsdateien im Ini-Format gesteuert und eignet sich deshalb nur für fortgeschrittene Anwender mit Kenntnissen im Druckbereich.**

Lizenzbedingungen

Nutzung:

Der Autor gewährt dem Anwender das Recht die Software („PrintMulti“) kostenlos auf einer beliebigen Anzahl von Rechnern unter einem Windows-Clientbetriebssystem (2000, XP, Vista, Windows 7) zu nutzen, auch falls dieses als Druckserver fungieren sollten.

Für die Nutzung auf Serverbetriebssystemen (2000/2003/2008 Server, Terminalserver, ...) ist eine kostenpflichtige Lizenz erforderlich. Beim Drucken erscheint auf der ausgedruckten oder gespeicherten Seite ein Lizenzhinweis.

Ausgenommen von der kostenpflichtigen Nutzung auf Servern ist die Verwendung der Logfunktionalität, die für alle Betriebssysteme frei ist.

Für den Erwerb von Serverlizenzen setzen Sie sich bitte mit uns in Verbindung.

Haftungsausschluss:

Die Benutzung der Software erfolgt auf eigene Gefahr. Die Software und Dokumentation wird wie sie ist zur Verfügung gestellt.

Mit dem Installieren oder Benutzen der Software erklären Sie sich damit einverstanden, dass der Autor unter keinen Umständen für Schäden haftbar gemacht werden kann, die direkt oder indirekt durch die Installation, den Einsatz der Software oder der Dokumentation aufgetreten sind.

Uneingeschränkt ausgeschlossen ist vor allem die Haftung für Schäden aus entgangenem Gewinn, Betriebsunterbrechung, Verlust von Informationen und Daten, Schäden an anderer Software, auch wenn diese dem Autor bekannt sein sollten.

Ausschließlich der Benutzer haftet für Folgen der Benutzung dieser Software.

Weitergabe

Das Programm darf zusammen mit allen zugehörigen Dateien außer Lizenzdaten und in unverändertem Zustand weitergegeben werden.

Weiterhin dürfen Sie *PrintMulti* ohne weitere Genehmigung mit Ihrer selbst entwickelten Software nutzen, verteilen und installieren, sofern sie die Nutzungsbedingungen einhalten.

Veröffentlichungen auf CD-ROM innerhalb von Sharewaresammlungen und Zeitschriften sind gestattet.

Manipulation, Dekompilierung und Disassemblierung der Software und mitgelieferten Dateien werden nach § 263a StGB mit Freiheitsstrafe von bis zu 5 Jahren oder Geldstrafe geahndet, vom Autor zur Anzeige gebracht und Schadensersatz geltend gemacht.

Anwendbares Recht

Auf diese Lizenzvereinbarung ist ausschließlich Deutsches Recht anwendbar, der Gerichtsstand ist Karlsruhe/Baden. Die Wahl eines anderen nationalen Rechts oder eines anderen Gerichtsstands ist ausgeschlossen.

Schlussbestimmungen

Sollte irgendein Teil dieser Lizenzvereinbarung unwirksam oder undurchführbar sein, wird dadurch die Wirksamkeit der übrigen Bestimmungen dieser Vereinbarung nicht berührt, sondern diese behalten weiterhin ihre Gültigkeit.

Änderungen

Version 1.0.4.0

Probleme mit der Option "*Append2File*" behoben.
Zweite unabhängige Logdatei für Jobs wurde nicht erzeugt.

Version 1.0.3.3

Neues Tool "*ShowPrinterInformation*"
Neue Aktion "*SaveEmf*"
Aktualisiertes Tool "*Devmode2File*" (lange Druckernamen)

Version 1.0.3.2

Verwendung von Schriftarten, die in Spooldateien eingebettet sind. Ist vor allem für den PDF-Druck notwendig

Version 1.0.3.0

Windows 7 und Server 2008 R2 Unterstützung.
neues Makro „**p**“ liefert den Standarddrucker des druckenden Benutzers. Sinnvoll für Terminalserver.
Neuer Operator „**rawpage**“ liefert eine Textrepräsentation der gedruckten Seite. Allerdings sind keine Formatierungen enthalten. Alle Texte werden ohne Trennzeichen direkt hintereinander ausgegeben.
Neuer Operator „**tblsearch**“ ermöglicht die Suche nach Begriffen in Texten. Insbesondere durch die Kombination mit „**rawpage**“ kann eine dokumentenabhängige Konfiguration ermöglicht werden.
Neuer Operator „**tbllookup**“ vereinfacht die Konfiguration bei mehreren festen Werten (z.B. Benutzernamen). Ersetzt verschachtelte Abfragen.
Neuer Wert „**WriteTextToFile**“ bei den „Print“-Aktionen. Ermöglicht ein Abspeichern des Seitenquelltextes (wie bei „**rawpage**“) in Dateien. Kann innerhalb von Scripten verwendet werden. Ein Beispiel ist vorhanden.
Neuer Wert „**RefreshConfig**“ in „Exec“- und „Print“-Aktionen. Bewirkt erneutes Lesen der Konfigurationsdatei nach dem Abschnitt. Ermöglicht dynamische Änderungen an der Konfigurationsdatei.
Script „**gscreatepdf.vbs**“ ersetzt durch „**gscreatepdf1.vbs**“ wegen Problemen bei Pfadnamen mit Leerstellen oder Sonderzeichen.

Version 1.0.2.1

Berücksichtigung der Einstellungen "**Collate, Color, nUP, DrvCopies, TotalCopies**" für den PrintMulti Drucker. Wird auch beim Ausdruck auf den Hauptdrucker verwendet, falls der Abschnitt nicht aktiv ist.

Technische Hintergründe, Einschränkungen

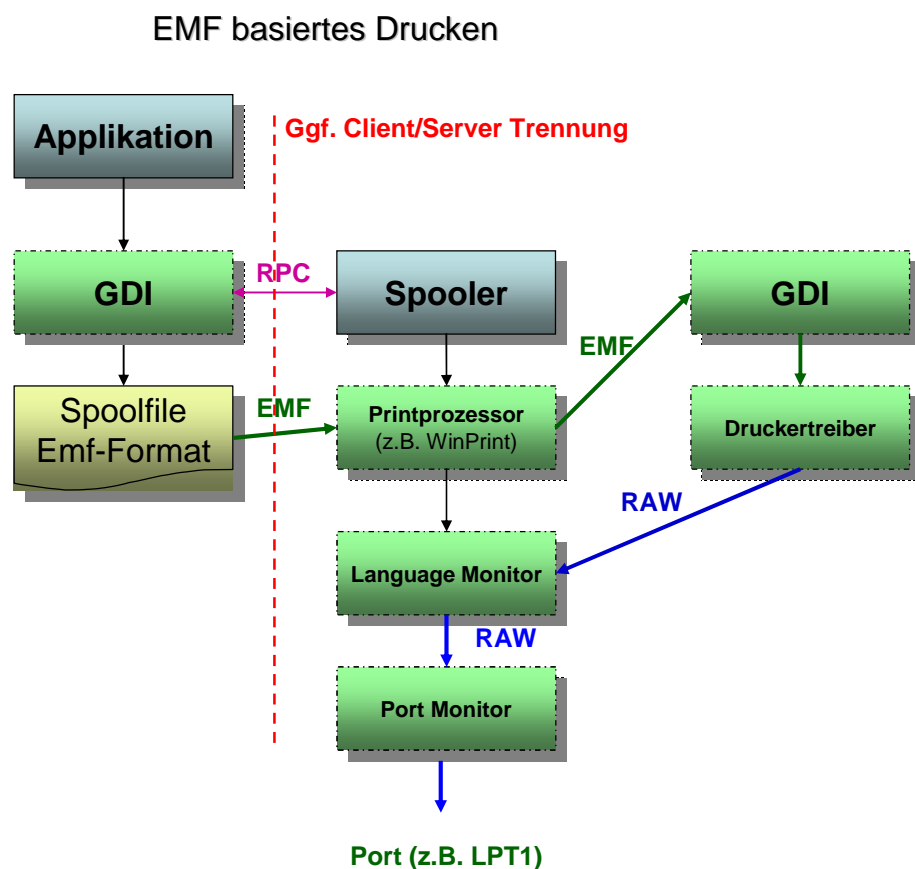
Technische Hintergründe

Ein Printprozessor ist eine DLL, die vom Windows Spooler verwendet wird um eine vom GDI aufgenommene Spooldatei abzuarbeiten. Der folgende Link zeigt eine gute Übersicht über die beim Drucken involvierten Komponenten.

<http://msdn2.microsoft.com/en-us/library/ms801074.aspx>

Im Folgenden wird der „normale“ EMF-Druckvorgang, der RAW-Druckmodus und der Mechanismus erläutert, den PrintMulti verwendet.

Der EMF-Druckvorgang



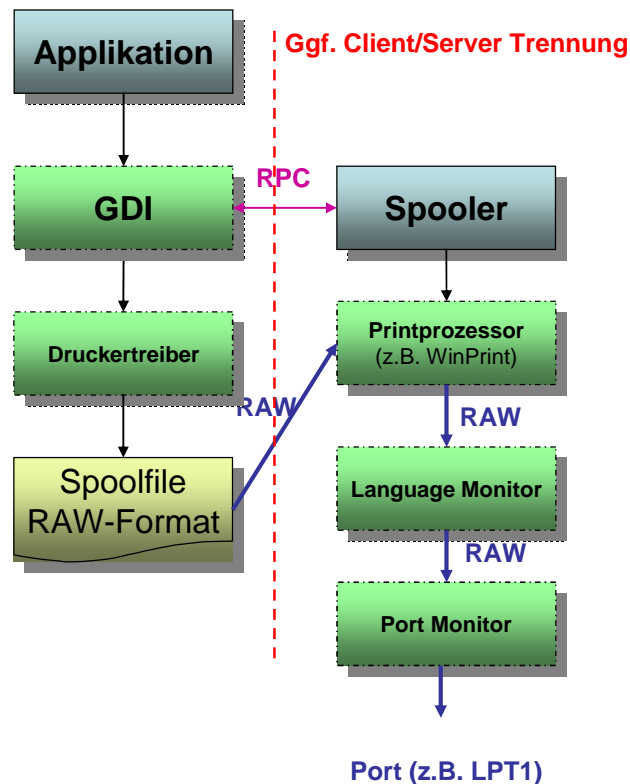
Eine druckende Anwendung ruft GDI Grafikbefehle auf, die in eine Spooldatei geschrieben werden (fast unverändert als EMF-Records). Der Spooler wird über einen RPC Aufruf mit der Verarbeitung des Druckjobs beauftragt. Falls auf einen Netzwerkdrucker ausgegeben wird, übernimmt der Spooler auf dem Server die Verarbeitung. Die Spooldatei wird dazu vorher auf den Server übertragen und ggf. angepasst (z.B. Fonts eingebettet, die auf dem Server fehlen). Der Spooler ruft jetzt festgelegte Einsprungpunkte des für den Drucker eingetragenen Printprozessors auf. Dieser nutzt das GDI um die EMF-Befehle über den Druckertreiber in die RAW-Daten für den jeweiligen Drucker zu übersetzen.

Durch die Verwendung des Spoolers ist das zeitraubende Rendering in die RAW-Daten vom Druckvorgang der Anwendung entkoppelt, wodurch die Anwendungen nach dem Drucken deutlich schneller wieder fortfahren können.

Dieser Druck-,„Pfad“ ist für einen reibungslosen Ablauf von PrintMulti unbedingt erforderlich.

Der RAW-Druckvorgang

RAW basiertes Drucken



Leider gibt es Situationen in denen der Rendervorgang bereits im ersten Schritt vom GDI angestoßen wird und der entsprechende Printprozessor dann die RAW-Daten, die auch in der Spooldatei gespeichert sind, zur Verarbeitung erhält. Da die Daten bereits im druckerspezifischen Zielformat vorliegen, ist eine Bearbeitung, wie PrintMulti sie vornimmt, nicht mehr möglich.

In PrintMulti ist momentan keine RAW-Verarbeitung vorgesehen. Nutzen Sie ggf. andere Tools (z.B. Redmon).

In folgenden Fällen wird kein EMF-Spooling verwendet:

(siehe auch: http://www.undocprint.org/winspool/spool_files#when_is_raw_used)

Druckaufträge werden direkt zum Drucker geleitet (Druckereigenschaft / Erweitert)

Die erweiterten Druckfunktionen sind nicht aktiviert (Druckereigenschaft / Erweitert und auch bei Geräteeinstellungen für den Drucker).

Einige Treiber drucken immer RAW (vor allem die von neueren Druckern von den Homepages der Druckerhersteller).

Die in Windows mitgelieferten funktionieren meist problemlos.

Bei der Verwendung von Netzwerkdruckern müssen Betriebssysteme, EMF-Versionen und auch Druckertreiberversionen passen (im letzteren Fall ggf. die Treiber von den Clients löschen und beim Verbinden mit dem Server neu installieren lassen. Dann sollten die Versionen gleich sein).

Der Adobe Reader druckt auf Druckern, die Postscript Treiber verwenden, immer RAW aus. Verwenden Sie also keinen Postscript-Treiber für PrintMulti, falls Sie gedenken mit dem Adobe Reader PDF Dateien zu drucken (oder nehmen einen anderen Viewer).

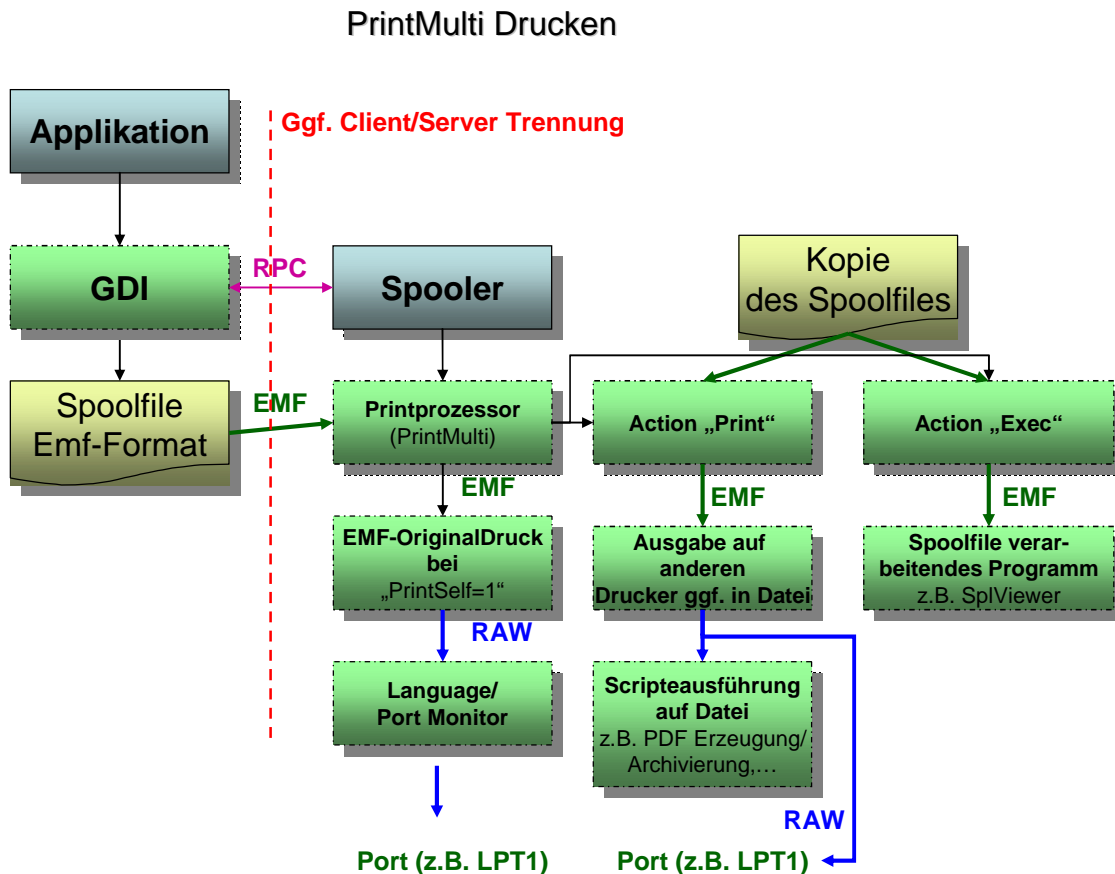
Die oben verwiesene Seite enthält noch einige zusätzliche Fälle.

Falls PrintMulti einen RAW-Job verarbeiten soll, dann wird dieser unverändert weitergegeben. Die Logdatei enthält dann folgenden Eintrag:

DEBUG ;...;;ATTENTION! PrintMulti does not support RAW printing. Job will go to the original printer without PrintMulti features.

Sie können den RAW Modus meist vermeiden, indem Sie einen Dummy-Drucker installieren (siehe Kapitel „Die Installation“)

PrintMulti-Drucken



PrintMulti kopiert die ursprüngliche Spooldatei und führt Aktionen darauf aus. Der Originaljob kann in vielen Fällen zusätzlich ausgeführt werden („PrintSelf=1“). Insbesondere bei der Verwendung von in Windows enthaltenen Treibern. Standardmäßig wird er aber unterdrückt.

Es gibt Aktionen, welche die kopierte Spooldatei durch andere Drucker leitet und dabei viele Einstellmöglichkeiten und Scriptfähigkeiten bietet („Print“-Aktionen).

Eine weitere Art von Aktionen baut auf Programmen auf, welche die Spooldatei direkt verarbeiten können. Hier ist vor allem der SplViewer zu nennen.

Einschränkungen und bekannte Probleme

Folgende Probleme sind bekannt:

In vielen Fällen wird ein RAW-Druckjob erzeugt. Siehe dazu den Abschnitt „Der RAW-Druckvorgang“.

➔ *Anderen Druckertreiber (z.B. HP Color Laserjet 8550 PCL) als PrintMulti-Drucker verwenden.*

Der Adobe Reader erzeugt beim Drucken teilweise temporäre Schriftarten, die beim Drucken innerhalb von PrintMulti schon gelöscht sein könnten und dann zu seltsamen Ausdrucken führen. Das Problem scheint abhängig von der Art des Font-Embeddings in der PDF Datei zu sein (die besten Einstellungen für die PDF Erzeugung mit Ghostscript werden später erläutert).

➔ *PDF konvertieren oder „Als Bild drucken“ verwenden (langsam!) oder einen anderen PDF-Viewer verwenden (z.B. FoxIt).*

Falls der Druckertreiber, den Sie für den PrintMulti-Drucker verwenden Probleme bereitet, dann verwenden Sie einen in Windows enthaltenen, am besten basierend auf dem Standard Printprozessor „WinPrint“.

➔ *Anderen Druckertreiber (z.B. HP Color Laserjet 8550 PCL) verwenden.*

PrintMulti basiert auf dem Kopieren und Ausführen einer EMF-Datei die für einen Drucker erzeugt wurde und auf einem anderen ausgegeben wurde. Das ist nur dann möglich, falls die EMF-Dateien geräteunabhängig sind. Leider ist das nicht immer der Fall.

Der gängige Fall, dass Excel beim Drucken von dünnen Linien auf PCL-Druckern RAW-Befehle in die EMF-Datei einbettet (ein uraltes Optimierungsrelikt), wird von PrintMulti abgefangen. Es gibt aber noch weitere Fälle die theoretisch möglich sind (z.B. Drucker, die kein StretchBlt unterstützen, etliche Postscript Optionen).

➔ *Anderen Druckertreiber (z.B. HP Color Laserjet 8550 PCL) verwenden und Kontakt mit uns aufnehmen.*

Die PDF Erzeugung (siehe Beispiel: "ActionCreatePDFScript") funktioniert unter Umständen auf einem freigegebenen PrintMulti-Drucker nicht für druckende Clients (lokale funktionieren).

Die Ursache ist, dass das Erzeugen der Datei mit der Option "Save2File" aus unbekanntem Gründen für die druckenden Clients nicht funktioniert, trotz ausreichender Rechte.

➔ *Mit der Einstellung "UseSystemAccount=1" versuchen.*

Installation

Die Installation von PrintMulti verläuft unspektakulär in wenigen Augenblicken ab, da nur einige Dateien kopiert werden. Der eigentliche PrintProcessor (die „printmulti.dll“) wird notwendigerweise ins Verzeichnis „%systemroot%\system32\spool\prtprocs\w32x86“ kopiert und die entsprechenden Registryeinträge erzeugt. Bei 64 Bit Betriebssystemen ist das Ziel „... \prtprocs\x64“.

Einige weitere Dateien (Handbücher, Skripte, Konfigurationsdatei „printmulti.ini“, ...) werden stets ins Verzeichnis „%programfiles%\printmulti“ kopiert. Für das Handbuch wird ein entsprechender Verweis im Startmenu von Windows eingerichtet.

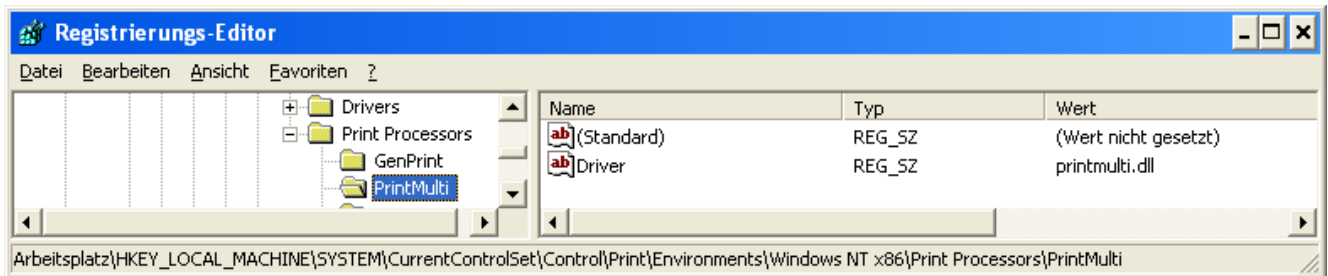
Zusätzlich ist auch noch ein ICC-Profil mit den entsprechenden Lizenzinformationen (von www.eci.org) für die PDF/A Erstellung enthalten. Es können auch weitere freie Farbprofile von dort heruntergeladen werden. Für Farbprofile ist von Windows der Pfad „%systemroot%\system32\spool\drivers\color“ vorgesehen. Im Script „gscreatepdf1.vbs“ wird darauf verwiesen.

Vorhandene Dateien werden bis auf die printmulti.dll nicht überschrieben. Bei einem Update wird die neue, unter Umständen geänderte Beispielkonfigurationsdatei als „*printmulti.sample.ini*“ gespeichert.

Bei einem Update von PrintMulti muss die Datei „printmulti.dll“ ersetzt werden. Dies erfordert normalerweise einen Neustart des Rechners. Dieser kann meist vermieden werden, indem vor dem Kopieren der Spooler mit dem Befehl „*net stop spooler*“ angehalten und nach der Installation mit „*net start spooler*“ wieder gestartet wird. Achten Sie darauf, dass keine Druckjobs gerade im Gange sind. Das Installationsprogramm weist Sie auf den richtigen Zeitpunkt zur Ausführung der Befehle hin.

Bei der Deinstallation wird alles bis auf das ggf. kopierte ICC Profil wieder entfernt. **Insbesondere wird der komplette Installationspfad gelöscht, auch falls sich dort zusätzliche Dateien befinden sollten.**

Falls Sie die Installation in Ihre Software integrieren wollen, genügt es die DLL ins entsprechende Printprocessorverzeichnis zu kopieren und die Api-Funktion „AddPrintProcessor()“ aufzurufen. Diese erzeugt folgende Einträge in der Registry:



Die Einträge können ggf. auch manuell hinzugefügt oder entfernt werden. Der Dienst „Druckerwarteschlange“ (der Spooler) muss dann aber neu gestartet oder der Rechner gebootet werden.

Die Installationsdatei enthält Versionen für 32 und für 64 Bit. Die zu installierende Version wird durch ein kurzes Startprogramm ermittelt und das entsprechende Installationsprogramm betriebssystemabhängig aufgerufen.

Hier eine Übersicht über die installierten Dateien und deren Bedeutung:

| Dateiname | Pfad | Bedeutung |
|----------------------------|-------------|--|
| Printmulti.dll | 1) | Die PrintProcessor-DLL |
| eciRGB_v2.icc | 2) | Einfaches Farbprofil für PDF/A Erzeugung von www.eci.org |
| eciRGB_licence.rtf | 2) | Lizenz für die obige Datei |
| printmulti.ini | 3) | Beispielkonfigurationsdatei die nach der Installation verwendet wird. |
| Printmulti_d.pdf | 3) | Dieses Handbuch |
| Uninst.exe | 3) | Aufruf des Uninstallers (auch über Systemsteuerung/Software möglich) |
| devmode2file.exe | 3) | Ermöglicht das Speichern von Druckereinstellungen in Dateien. |
| ShowPrinterInformation.exe | 3) | Zeigt (und speichert) Informationen über Drucker |
| gscreatepdf1.vbs | 4) | Script für die PDF Erzeugung mit Ghostscript (Ersetzt die „gscreatepdf.vbs“, die bei großen Postscriptdateien zu Problemen mit Ghostscript führte) |
| PDFA_def.ps | 4) | Postscriptdatei zur Erzeugung von PDF/A Dokumenten (enthält Macros) |
| CmdRedirect.exe | 4) | Nutzen von Kommandozeilenparametern für die Umleitung der Standardausgabe für den Aufruf von Ghostscript im GSCreatePDF1.vbs Script. |
| TextExtract.vbs | 4) | Zeigt wie nach Daten innerhalb von gedruckten Texten gesucht werden kann |
| textextractsample.doc | 4) | Enthält versteckte Informationen, die von TextExtract.vbs extrahiert werden. |
| SendMailExample.doc | 4) | Beispiel zum E-Mailversand mit enthaltener Zieladresse |
| Mailbody.txt | 4) | Hilfsdatei mit dem Nachrichtentext für den E-Mailversand |
| Blat.exe | 4) | Freies Kommandozeilenprogramm zum E-Mailversand |

Referenzierte Pfade (auch weiter unten):

- 1) %systemroot%\system32\spool\prtprocs\w32x86
- 2) %systemroot%\system32\spool\drivers\color
- 3) Installationsverzeichnis von PrintMulti (%programfiles%\printmulti)
- 4) Verzeichnis (%programfiles%\printmulti\PDFCreation)

Folgendes sollten Sie zusätzlich installieren bzw. kopieren:

Falls Sie direkt mit Ghostscript PDF Dateien erzeugen wollen, sollten Sie Ghostscript ab Version 8.61 installieren (Versionen davor erzeugen keine 100% konformen PDF/A Dokumente). Das Script erkennt automatisch den Installationspfad von Ghostscript.

<http://downloads.sourceforge.net/ghostscript/g861w32.exe>

Für eine Volltextextraktion aus der PDF-Datei kann „pdftotext“ verwendet werden. Am Ende der „gscreatepdf1.vbs“ ist ein Beispiel enthalten. Sie müssen dazu die „pdftotext.exe“ ins Verzeichnis printmulti-Verzeichnis ⁴⁾ kopieren.

<http://www.foolabs.com/xpdf/download.html>

Sie sollten vielleicht noch wissen, dass der Installationspfad von PrintMulti in der Registry unter *„HKEY_LOCAL_MACHINE\SOFTWARE\LVBPrint\PrintMulti\InstallPath“*

abgelegt und Scripten und der PrintMulti.ini über die Umgebungsvariable *“PM_INSTALLPATH”* zur Verfügung gestellt wird.

Drucken mit PrintMulti

Alle relevanten Daten werden aus einer Konfigurationsdatei geladen, deren Speicherort in der Registry im Key „*HKEY_LOCAL_MACHINE\SOFTWARE\LVBPrint\PrintMulti*“, „*ConfigurationFile*“ hinterlegt ist. Falls dort kein Pfad zu finden ist, wird im Windows-Pfad nach der Datei gesucht.

Bei der Installation wird eine Beispieldatei im „<Programmpfad>\Printmulti“ angelegt und der entsprechende Registryverweis erzeugt.

Um die Möglichkeiten von PrintMulti zu nutzen, installieren Sie am besten einen neuen lokalen Drucker, der in eine feste Datei druckt. Ab Version 1.0.1.3 ist es auch möglich einen bestehenden Drucker zu verwenden. Sie können dann wählen, ob der Druckjob auf diesem Drucker tatsächlich ausgegeben wird oder nur auf den/die zusätzlich konfigurierten Ausgabewege. Wie bereits mehrfach erwähnt, funktioniert das aber meist nur mit Druckertreibern, die bei Windows mitgeliefert sind.

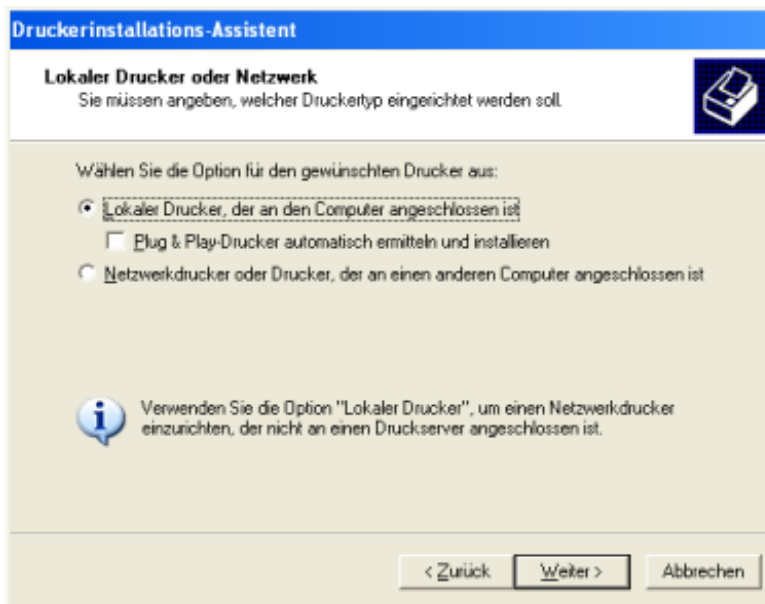
Folgende Bedingungen müssen zutreffen, damit PrintMulti für einen Drucker in Aktion tritt:

1. Dem Drucker muss in den erweiterten Einstellungen als PrintProzessor „*PrintMulti*“ gewählt werden.
2. In der relevanten Konfigurationsdatei muss es einen Abschnitt mit dem Drucker geben, der Wert „*Active*“ muss in diesem Abschnitt auf „1“ gesetzt und mindestens eine durchzuführende Aktion konfiguriert worden sein.

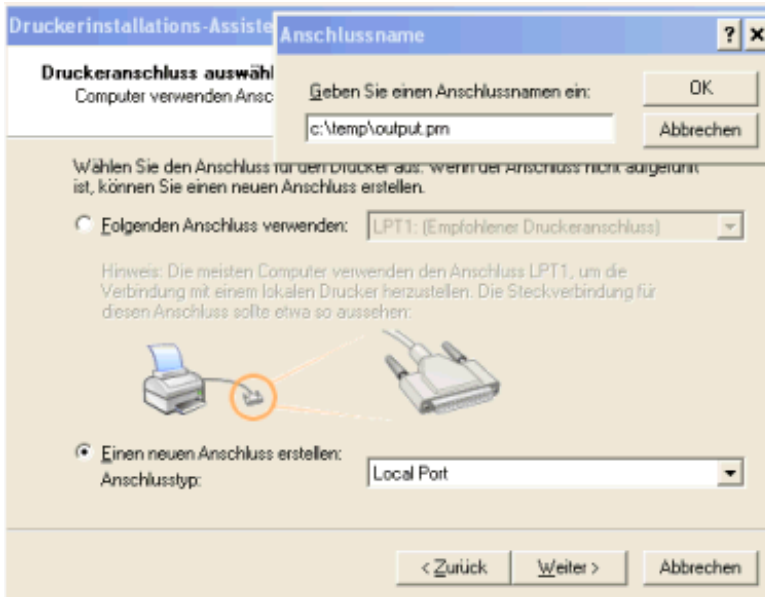
Bei Problemen überprüfen Sie zunächst diese Bedingungen.

Im Folgenden seien die Einstellungen für einen neuen Drucker „*Fotoabzug*“ dargestellt, der alle gedruckten Seiten auf einem Drucker „*Fotoprinter*“ einmal unverändert und am Ende nochmals 9 Seiten pro Blatt in Schwarzweiß druckt.

Einige Zwischenschritte sind ausgelassen.

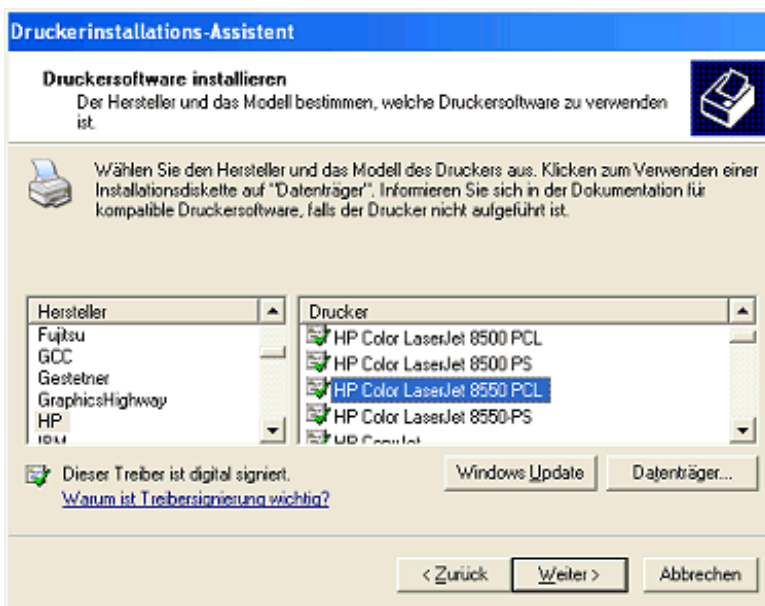


Neuen lokalen Drucker einrichten.



Als Port z.B. einen neuen „Local Port“ mit einem Dateinamen als Ziel verwenden (Falls Sie den Drucker freigeben, müssen alle Benutzer Schreibrechte auf die Datei haben!)

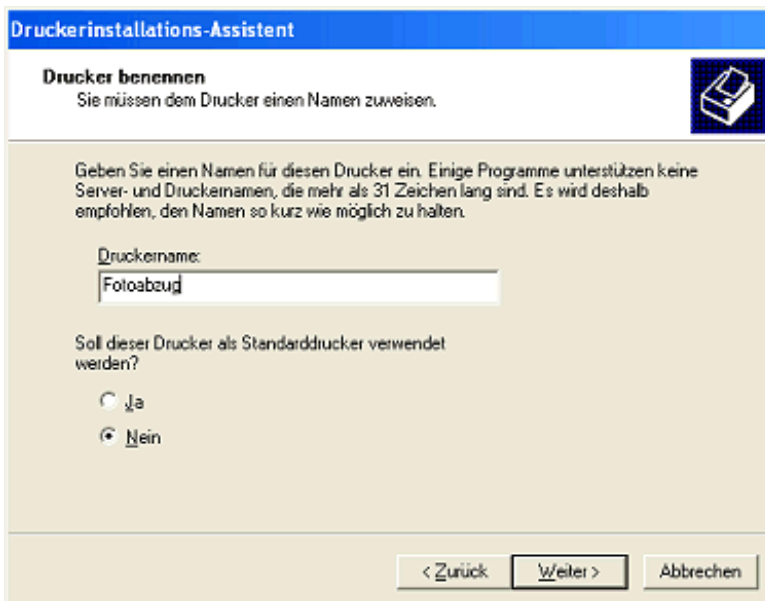
Nachdem dieser Drucker mit „PrintMulti“ eingerichtet ist, erfolgt keine physikalische Ausgabe mehr auf das Gerät.

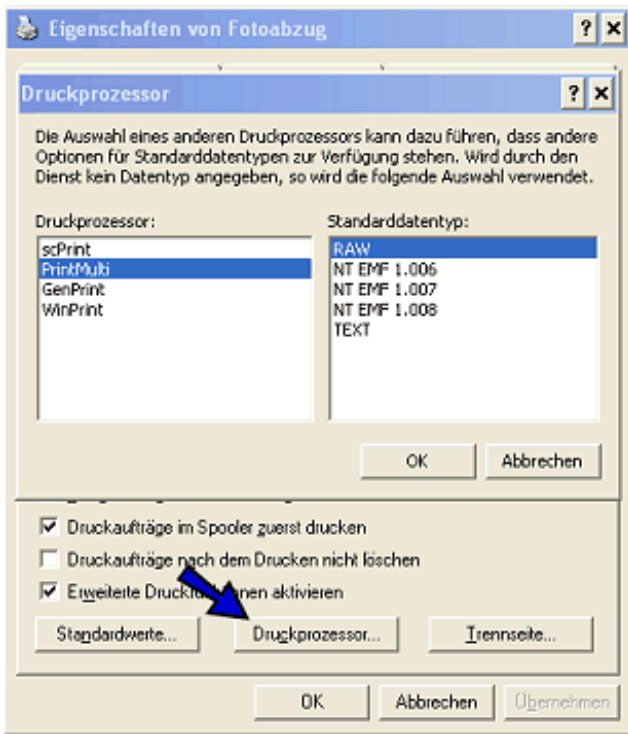


Wähle einen Drucker, der Din A3-Druck, Farbe und Duplex beherrscht.

Wahrscheinlich ist es am sinnvollsten, einen Drucker des gleichen Herstellers wie des physikalischen Druckers zu verwenden.

Vermeiden Sie Postscript Treiber, falls Sie mit dem Acrobat Reader ausdrucken möchten (→RAW-Druck).





Nicht vergessen bei den installierbaren Optionen für den Drucker die Duplexeinheit zu aktivieren, falls Sie den Duplexdruck wünschen.

„PrintMulti“ kann jetzt als Druckprozessor eingestellt werden. Der Datentyp kann auf RAW bleiben.

[Common]

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

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

[Fotoabzug]

```
Active=1
PrintSelf=0
Action1=Print;ActionFotodrucker1Co
Action2=Print;ActionFotodrucker9BW
```

[ActionFotodrucker1Co]

```
Active=1
Printer=Fotodrucker
nUp=1
```

[ActionFotodrucker9BW]

```
Active=1
Printer=Fotodrucker
Color=0
nUp=9
nUpBorder=1
```

Minimale Konfigurationsdatei:

„C:\Programme\PrintMulti\printmulti.ini“

Es werden zwei Logdateien geschrieben - eine mit den Jobdaten und eine mit ausführlichen Debuginformationen.

Der Druckjob auf den Drucker „Fotoabzug“ wird an die beiden Druckaktionen in der angegebenen Reihenfolge „gesendet“, falls „Active“ auf 1 gesetzt ist. Auf den Drucker „Fotoabzug“ erfolgt standardmäßig keine Ausgabe. Nur falls „PrintSelf“=1, wird die Originalausgabe (mit allen Einstellungen) durchgeschleust.

Anderenfalls wird herkömmlich ausgedruckt, so als wäre der Drucker mit dem Druckprozessor „winprint“ verbunden. Die Logdaten werden dennoch erzeugt.

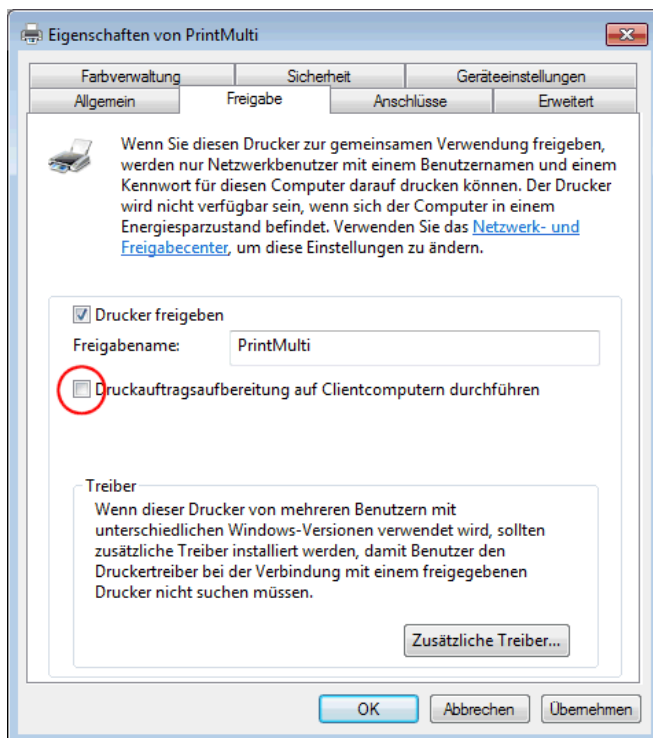
Freigeben von PrintMulti-Druckern

Wie bereits mehrfach erwähnt, basiert PrintMulti darauf, dass die Druckjobs im EMF-Format "geliefert" werden. Gerade in Client/Server Umgebungen hat Microsoft die bevorzugte Architektur geändert und propagiert nun das "Client-Side Rendering" (CSR) (<http://blogs.technet.com/b/askperf/archive/2008/02/10/ws2008-client-side-rendering.aspx>)

Durch das CSR wird der Druckjob auf dem Client gerendert und im entsprechenden RAW-Format des Druckers (z.B.: PCL) zum Server gesendet.

Das CSR wird bei der Freigabe eines Druckers gesetzt. Es gibt aber laut dem obigen Artikel wohl einige Konstellationen, wo unabhängig von der Freigabeeinstellung, CSR benutzt wird, obwohl ausgeschaltet bzw. CSR nicht benutzt wird, obwohl eingeschaltet.

Bei der Freigabe eines Druckers unter Vista, Windows7, Server 2008 ist deshalb die entsprechende Einstellung zu deaktivieren:



Falls Sie sich mit Windows 7 Clients auf ältere Server (2003) oder Clients (XP) verbinden, dann müssen Sie auf dem Client in den Eigenschaften des verbundenen Druckers die CSR Eigenschaft setzen. Der Server bietet diese Einstellmöglichkeit im Freigabedialog nicht, obgleich der Client den Wert zu verwenden scheint.

Achtung!

Es scheint teilweise zu Problemen beim Speichern von Dateien (Option "Save2File") zu geben. Dies wird z.B. beim Erzeugen von PDF-Dateien mit Ghostscript ("ActionCreatePDFScript") verwendet.

Falls derartige Probleme auftreten sollten, dann versuchen Sie es mit der Einstellung "UseSystemAccount=1" im entsprechenden Abschnitt.

Sie können auch 64 Bit Clients mit 32 Bit Servern und andersherum verwenden. Das funktioniert gut, falls Sie einen Treiber verwenden, der auf den UNIDRV-Treibern von Microsoft basiert und Sie die Treiber auf dem Server im Freigabedialog (siehe oben "Zusätzliche Treiber") zur Verfügung stellen.

Es werden einige Dateien vom entsprechenden Client benötigt (stdnames.gpd, ttfsub.gpd, unidrv.dll, unidrv.hlp, unidrvui.dll, unires.dll). Die DLL Typen müssen passen (falls Sie einen 64 Bit Treiber hinzufügen, müssen die DLLs auch für 64 Bit kompiliert worden sein).

Expandierung von Makros und bedingten Ausdrücken

Alle Einstellungen in der Konfigurationsdatei können festgelegte Makros und berechenbare Ausdrücke, basierend auf einer einfacher Stackmaschine (wie alte HP-Taschenrechner) enthalten.

Durch diese Neuerung wird PrintMulti unheimlich flexibel. Lesen Sie später bei den Beispielen, wie Sie z.B. folgendes erreichen können:

Papierformatbasiertes Drucken (auf einen Drucker A3 Jobs, auf einen anderen A4 Jobs)

Benutzerabhängiges Drucken (z.B. nur bestimmte Personen dürfen farbig drucken)

Seitenzahlabhängiges Drucken (z.B. ab 1000 Seiten automatisch auf einen schnelleren Drucker oder in diesem Fall Duplex mit 4 Seiten/Sheet)

Dokumentabhängiges Drucken (z.B. Dokumentenname enthält „Rechnung“ dann auf einen hochwertigen Drucker)

Datumsabhängiges Drucken (z.B. Nachts auf einen leisen Drucker)

Der unbedarfte Anwender könnte trotzdem nur einen Drucker „sehen“.

Makros

Makros sind formatierbare Kürzel für bekannte Daten aus dem Druckjob, der Konfiguration oder von Umgebungsvariablen. Der oft angesprochene „Devicemode“ enthält Einstellungen, die der Anwender beim Drucken in den Druckereinstellungen vorgenommen hat wie das Papierformat, den Duplexmode (Drucken auf Vor- und Rückseite), ...

Der Devicemode enthält viele Standardeinstellungen, die als Makro zur Verfügung stehen und zusätzliche druckertreiberabhängige Werte auf die nicht zugegriffen werden kann (z.B. Ausgabeschächte, ...)

Hier ein Beispiel aus der Konfiguration der Logdatei, welche viele Beispiele enthält.

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

Es gelten folgende Regeln für die zu ersetzenden Strings:

Ein Makro beginnt mit ‚#‘ wahlweise gefolgt von einem Formatausdruck in Klammern und einem abschließenden Buchstaben z.B.: „#(...)Z“ oder

ein durch ‚%‘-Zeichen eingeschlossene Umgebungsvariable. Z.B.: „%TEMP%“. Es können nur Systemumgebungsvariablen verwendet werden.

Hier zunächst die verschiedenen Makrokürzel

| Makro | Type | Bedeutung |
|--|--------|--|
| Sonstige und Informationen aus den Jobdaten des Druckjobs | | |
| A | String | Name der Aktion (z.B. „ActionFotodrucker1Co“ im Beispiel oben) |
| B | String | Aktionstyp, momentan „print“ oder „exec“ |
| C | DWORD | Druckerspezifischer Zähler (für den Hauptdrucker). Wird mit jedem Druckjob hochgezählt und ist in der Registry beim Drucker hinterlegt. |
| D | String | Name des Dokuments, wie in die Anwendung übergibt |
| K | String | Name des Dokuments wie bei #D. Ungültige Zeichen für Verzeichnisnamen werden durch einen Unterstrich ersetzt. (<>:“^) |
| E | String | Aktionsspezifischer Text wird ersetzt durch den Druckernamen des Slavedruckers bei Druckausgaben (z.B. „Fotodrucker“) |
| F | String | Dateiname der kopierten Spooldatei |
| G | String | Aktionsspezifischer Dateiname. Falls bei einer „print“-Aktion in eine Datei gedruckt wird, in das dieser Dateiname. In allen anderen Fällen der Name |

| | | |
|---|-----------|---|
| | | der (kopierten) Spooldatei. (z.B. „c:\temp\file.pcl“) |
| g | String | Wie bei „G“, nur ohne Dateiendung (z.B. „c:\temp\file“) |
| I | String | Auswerten des Ausdrucks aus einem andere Konfigurationsabschnitt („Include“) |
| J | DWORD | JobId des Druckjobs (nicht eindeutig, wird von Windows wieder verwendet) |
| M | String | Computername des druckenden Clients |
| P | String | Name des Hauptdruckers (z.B. „Fotoabzug“) |
| p | String | Neu: Standarddrucker des druckenden Benutzers (funktioniert für freigegebene Drucker u.U. nicht) |
| S | Date/Time | Zeitpunkt der Druckjoberzeugung aus den Jobdaten |
| T | Date/Time | Aktuelle Zeit |
| U | String | Benutzername des druckenden Benutzers |
| Z | DWORD | Anzahl von Seiten aus den Druckjobdaten |
| z | DWORD | Gesamtanzahl von Kopien (nur für Action „Print“) |
| Daten aus dem Devicemode des Druckjobs | | |
| c | 0/1 | Farbe, 0 = sw, 1 = Farbe (dmColor – 1) |
| b | DWORD | Papierfach (dmDefaultSource) |
| d | 0/1/2/3 | 0 = nicht gesetzt, 1 = Simplex, 2 = Duplex vertikal, 3 = Duplex horizontal (dmDuplex) |
| o | 0/1/2 | 0=nicht gesetzt, 1 = Portrait, 2 = Landscape (dmOrientation) |
| s | DWORD | Papiergröße (z.B. 8 = A3, 9 = A4) (dmPaperSize), 0 = nicht gesetzt |
| y | DWORD | Auflösung (dmYResolution), 0 = nicht gesetzt |

Der erlaubte Formatausdruck hängt vom Typ ab. Folgende Formate sind für die unterschiedlichen Typen anwendbar:

String:

Beispiel: „meinTest“

Standard: vorhandener String, z.B. „#D“ „meinTest“

Formatanweisung: Eingebettet in sprintf mit „%<Format>s“, z.B. „#(-10)D“ gibt „%-10s“, im Beispiel „meinTest “, „,%“ oder „,*“ sind verboten im String und führen zu einer Standardformatierung.

DWORD oder Zahl:

Beispiel: 40000

Standard: „,%u“ z.B. „#C“ „,40000“

Formatanweisung: Falls das letzte Zeichen im Formatstring eines von „diuxXo“ ist, dann wird dieses zur Formatierung verwendet, ansonsten „,%u“. „#(08X)C“ wird zu „%08X“ im Beispiel zu „,00009C40“, „#(08)C“ zu „,%08u“ also zu „,00040000“. „,%“ im String ist verboten und führt zu einer Standardformatierung.

Datum / Zeit:

Beispiel: 23.11.1964 13:45:12,500

Standard: aus „printmulti.ini“ oder "%Y-%m-%d %H:%M:%S", falls nicht konfiguriert
Formatanweisung: wird mit der Funktion „strftime“ formatiert.. Bitte im Internet nachlesen.

Berechnete Ausdrücke

Allgemeine Regeln

Jeder Konfigurationseintrag kann auf der Datenseite Berechnungsausdrücke enthalten, die bei jedem Zugriff erneut evaluiert werden. Auf andere Einträge kann mit dem neuen Makro „# (<Section.Key>) I“ zugegriffen werden, wodurch sich die Übersichtlichkeit erhöht.

Es gelten folgende Regeln:

Ein Berechnungsausdruck hat das Format: „\$ (<Ausdruck>) “, wobei Ausdruck als „<Operand/Operator>;<Operand/Operator>;...;<Operand/Operator>“ definiert ist.

Ein Ausdruck besteht aus Operanden und Operatoren, die durch einen Strichpunkt voneinander getrennt sind. Operanden werden auf den Stack geschoben. Operatoren holen die entsprechende Anzahl von Operanden vom Stack, berechnen den Wert und schieben diesen wieder auf den Stack.

Am Ende der Berechnung muss sich genau ein Wert auf dem Stack befinden, sonst wird ein Fehler gemeldet.

Operanden müssen durch Anführungszeichen eingeschlossen sein, falls diese Operatoren oder Sonderzeichen, wie Strichpunkt oder die geschlossene Klammer enthalten.

Falls Operanden nicht durch Anführungszeichen eingeschlossen sind, werden Leerstellen am Anfang und am Ende entfernt.

Ein Operand wird als Zahl betrachtet, falls er nur Ziffern enthält (negative Zahlen sind vorerst nicht vorgesehen). Viele Operatoren haben unterschiedliche Bedeutung für Zahlen und Texte. Nur falls alle Operanden Zahlen sind, dann wird eine arithmetische Operation ausgeführt (z.B.: „\$(2;3;+)“ gibt „5“, „\$(2;A;+)“ gibt „2A“)

Nutzen Sie das Flag mit dem Wert „32“ im Common-Abschnitt der PrintMulti.ini um Debugmeldungen über Berechnungsausdrücke zu protokollieren.

Ein Ausdruck kann Makros enthalten, die für den Konfigurationseintrag gültig sind. (z.B.: „LastPage=\$ (#Z; 1; -)“, setzt als letzte zu druckende Seite die Anzahl von Seiten minus 1; druckt also die letzte Seite nicht aus)

Jeder Ausdruck wird bei der Verwendung erneut berechnet und kein gespeicherter Wert verwendet.

Bei Textoperatoren (and,or,not,contains,upper,lower,tbllookup,tblsearch) wird nicht zwischen Groß- und Kleinschreibung unterschieden (“AnD” wird auch erkannt)

Falls Sie den Seitentext verwenden (mit „RAWPAGE“) nutzen Sie die Option „WriteTextToFile“, falls etwas nicht klappt. Es funktioniert nicht bei allen druckenden Anwendungen und die zu suchenden Passagen müssen meist innerhalb der sichtbaren Ausgabe liegen und dürfen beim Drucken nicht abgeschnitten sein.

Tabelle von Operatoren

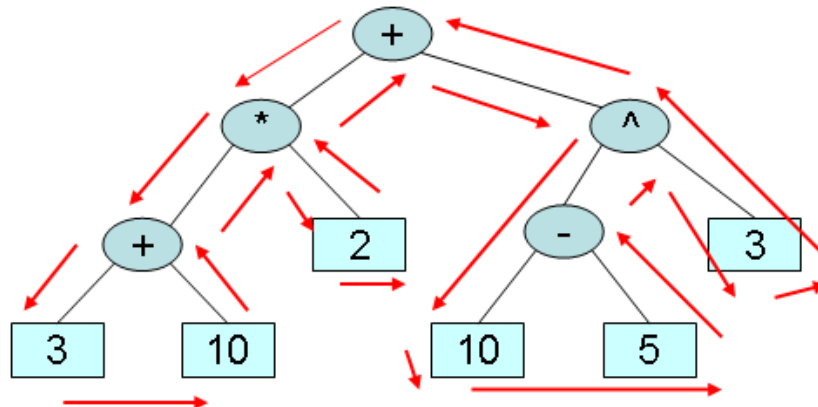
| Op | Beschreibung | 1) | Beispiel |
|-----------|---|-----------------------------------|---|
| + | Arithmetische Addition, falls beides Zahlen sind, sonst Stringverknüpfung | S | <code>\$(2;3;+)</code> 5 <code>\$(abc;4;+)</code> abc4 |
| - | Subtraktion für Zahlen | | <code>\$(100;5;-)</code> 95 |
| * | Multiplikation für Zahlen | | <code>\$(100;5;*)</code> 500 |
| / | Division für Zahlen, kein Runden | | <code>\$(101;3;/)</code> 33 |
| < | Falls beide zu vergleichenden Operanden Zahlen sind, dann wird ein arithmetischer Vergleich durchgeführt. Anderenfalls ein Stringvergleich ohne Berücksichtigung einer lokalen, also rein nach dem Unicodewert (siehe „wscmp“). Für die letzten beiden Operatoren gibt es jeweils zwei gleichwertige Kürzel. | S | <code>\$(A;B;<)</code> 1 |
| > | | S | <code>\$(A;a;>)</code> 0 |
| <= | | S | <code>\$(" Beta"; Alpha;<)</code> 1 <code>\$(" Beta" < "Alpha")</code> 1 |
| >= | | S | |
| ==,= | | S | <code>\$(A;A;==)</code> (1 |
| <>,! = | | S | <code>\$(" A"; A;! =)</code> (1 |
| !, not | | Liefert 1, falls der Wert = 0 ist | |
| and | Logische Operatoren benötigen zwei Zahlausdrücke. Es wird auf „!= 0“ verglichen. | | <code>\$(1;2;And)</code> 1 |
| or | | | <code>\$(0;0;or)</code> 0 <code>\$(9;0;or)</code> 1 |
| upper | Umwandlung in Groß- bzw. Kleinschreibung mit Verwendung der aktuellen lokalen Einstellung | L | <code>\$("äöüß"; upper)</code> <u>Logdatei:</u> |
| lower | | L | <code>Upper ("äöüß") ->"ÄÖÜß"</code> <code>Locale:German</code> |
| contains | „1“, falls der zweite String im ersten enthalten ist. „Case sensitiver“ Vergleich | S | <code>\$(ABCDEF;cd;contains)</code> 0 <code>\$(abCdE;upper;cd;upper;contains)</code> (1 |
| ? | Der C-Conditionaloperator; <Condition> ? <True-Value> : <False-Value> <code>\$(<Condition>;<True-Value>;<False-Value>;?)</code> | | <code>\$(A;B;<;1;2;?)</code> 1 Siehe spätere Beispiele |
| rawpage | Bekommt optional die Seitennummer als Parameter und liefert den Text des gedruckten Dokuments als String zurück (ohne Trennzeichen – direkt hintereinander). Ohne Angabe der Seitennummer werden alle Seiten, getrennt durch CR/LF zurückgegeben. | S | <code>\$(1;rawpage)</code> Seitentext der ersten Seite <code>\$(;rawpage)</code> alle Seiten Weitere Beispiele weiter hinten |
| tbllookup | Durchsucht alle Schlüssel in einem Abschnitt einer Konfigurationsdatei nach einem bestimmten Eintrag und gibt den zugewiesenen Wert oder den Default-Eintrag zurück, falls der Wert nicht gefunden wurde. Die Suche erfolgt unabhängig von Groß/Kleinschreibung. Falls „<File>“ leer, wird die aktuelle Konfigurationsdatei verwendet. Dateien ohne Pfadangabe werden im System32-Pfad gesucht. Verwenden Sie ggf. %PM_INSTALLPATH% als Pfad. <code>\$(<File>;<Section>;<Key>;<Default>;tbllookup</code> | S | <code>[Users]</code> <code>adam=\$(#Z;100;<)</code> <code>gustav=1</code> <code>[ActionPrint]</code> ... <code>Color=\$(;Users;#U;0;tbllookup)</code> ... Gustav druckt in Farbe; Adam nur für Dokumente mit weniger als 100 Seiten |

| | | |
|-----------|---|--|
| tblsearch | <p>Umgekehrte Suchweise wie bei „tbllookup“. Alle Einträge in einem Abschnitt werden der Reihe nach in einem String gesucht. Falls ein Eintrag enthalten ist (Groß/Kleinschreibung wird im Gegensatz zu „tbllookup“ beachtet!) wird der Wert ausgewertet und zurückgegeben. Die Suche wird beendet. Für den Dateinamen gilt entsprechendes wie oben.</p> <p><code>\$(<File>;<Section>;<String>;<Default>;tblsearch</code></p> | <p>S</p> <pre>[Printers] #invoice#=Printer1 #delivery#=Printer2 [ActionPrint] ... Printer=\$(;Printers;1;RAWPAGE; #p;tblsearch ... </pre> <p>Ein Worddokument könnte auf der ersten Seite („1;RAWPAGE“) unsichtbar für Rechnungsdruck irgendwo „#invoice#“ enthalten. Falls keiner der beiden Texte gefunden wird, dann erfolgt eine Ausgabe auf den Standarddrucker des Benutzers (#p).</p> |
|-----------|---|--|

- 1) S Stringrelevante Funktion
L Nutzt aktuelle Regionseinstellungen von Windows

Beispielumwandlung eines Ausdrucks in einen stackbasierten Ausdruck

$$(3 + 10) * 2 + (10 - 5) ^ 3$$



Beim intuitiven Parsebaum werden zunächst die linken Kinder betrachtet, dann die rechten und anschließend der Vaterknoten notiert. Die roten Pfeile verdeutlichen die Reihenfolge in der die Knoten durchsucht werden. Für das Beispiel würde sich der Ausdruck folgendermaßen als Stackausdruck darstellen:

3 10 + 2 * 10 5 - 3 ^ +

Beispiele zu diesem Themenbereich finden Sie weiter hinten bei den allgemeinen Beispielen.

Bereiche und Aktionen in der „printmulti.ini“

Die Konfigurationsdatei wird mit jedem Druckjob neu eingelesen. Änderungen wirken sich also beim nächsten Job direkt aus.

Für alle Werte mit dem Type „Bool“ wird „1“ und alle Texte, die mit „j“, „J“, „y“ oder „Y“ beginnen als „wahr“ erkannt. „Falsch“ wird bei „0“ oder Texten beginnend mit „n“ oder „N“ angenommen.

Common

Der Common-Abschnitt dient hauptsächlich dazu, Einstellungen für die Protokollausgabe festzulegen. Es können zwei getrennte Dateien angelegt werden, deren Dateiname auch Makros aus dem vorigen Abschnitt enthalten kann (Ausnahme: Devicemode-Macros). Durch die Stringersetzung des Dateinamens können vielfältige Logdateien erzeugt werden.

Z.B. „C:\LogFiles\#P\#U_#(%Y_%m)S.csv“ erzeugt für jeden Drucker und jeden Benutzer jeden Monat eine neue Datei hier z.B. „C:\LogFiles\FotoAbzug\dieter_2006_11.csv“.

Die abzuspeichernden Werte werden durch Macrostrings festgelegt. Alle Zeichen, die nicht zu einem Macro gehören, werden unverändert übernommen. Zusätzlich kann noch das Datumsformat festgelegt werden, das mit den formatlosen Macros „#T“ und „#S“ gespeichert wird.

Für die drei Ausgabemasken gelten folgende Bitfelder. Sie müssen die entsprechenden Werten zusammenaddieren um den gewünschten Wert zu erhalten (z.B. 27 = Mehr Info, Info, Warnungen und Fehler)

| | |
|----|--------------------|
| 0 | Keine Ausgabe |
| 1 | Fehler |
| 2 | Warnungen |
| 4 | Jobinfo |
| 8 | Info |
| 16 | Mehr Info |
| 32 | Berechnungsdetails |

Beispiel:

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

; two separate 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
```

Die mit „ „ gekennzeichneten Zeilen sind in einer Zeile und werden nur hier umgebrochen.

(Die LogJob... und LogFile... Einträge haben nur unterschiedliche Namen. Sie kennzeichnen einfach die Möglichkeit zwei verschiedene Logdateien mitzuschreiben. In der Tabelle ist nur eine Art aufgeführt. Die unterschiedlichen Standardwerte für die Formate sind im Beispiel ersichtlich)

Folgende Einstellungen sind verfügbar:

| Schlüssel | Standard | Bedeutung |
|------------------|----------------|--|
| DbgOutMask | 0 | Bitmaske zur Ausgabe von Meldungen für angeschlossene Debugger oder DbgView von Sysinternals |
| LogJobMask | 0 | Bitmaske zur Ausgabe der Meldungstypen |
| LogJobFile | “ | Ausgabedatei. Ohne einen gültigen Dateinamen (der Macros enthalten kann) wird nichts ausgegeben !! |
| LogJobFileFormat | Siehe Beispiel | Format für Debugausgaben |
| LogJobHeader | Siehe Beispiel | Ein Header, der u.U. bei neu angelegten Dateien eingefügt wird (vgl. nächster Eintrag) |
| LogJobHeaderOut | True | Header ausgeben bei neu erzeugten Logdateien |
| LogStdDateFormat | Siehe Beispiel | Datums/Zeitformat, das genutzt wird, falls bei Zeitformatierungen kein extra Format angegeben. |

Nach dem konfigurierbaren Abschnitt wird noch bei Debugausgaben die Meldung selber in die gleiche Zeile angehängt.

Für Jobausgaben wird zunächst noch der Status „FAILED“ oder „OK“ ausgegeben, gefolgt von der Aktion „Print“, „Exec“, ... anschließend der Name der Aktion und der Dauer der Aktion in ms.

Für den Hauptdruckjob wird ebenfalls nach allen Aktionen ein Jobeintrag erstellt. Dieser enthält „OK“, falls alle Aktionen erfolgreich waren; die Aktion und der Name der Aktion sind leer.

Einträge für den PrintMulti-Drucker

| Einstellungen für den PrintMulti-Drucker | | | |
|---|--------|---------|--|
| Einstellung | Type | Default | Bedeutung |
| Active | Bool | 0 | PrintMulti überhaupt verwenden? |
| PrintSelf | Bool | 0 | Ausgabe auf dem Drucker ausgeben (1) oder nur die Aktionen ausführen (0) |
| Action* | String | - | Aktionen in der angegebenen Reihenfolge ausführen. |
| Collate | Bool | - | Sortiereinstellungen für den Originaldruck verändern |
| Color | Bool | - | Farbeinstellungen für den Originaldruck verändern |
| nUp | Int | - | Anzahl Seiten pro Druckseite verändern |
| nUpBorder | Bool | - | Rahmen um die Einzelseiten ? |
| DrvCopies | Int | - | Anzahl von Kopien für den Treiber verändern. |
| TotalCopies | Int | - | Gesamtanzahl Kopien |

Wie bereits weiter oben beschrieben, muss für einen „Printmulti“-fähigen Drucker der PrintProzessor auf „***PrintMulti***“ festgelegt, ein Abschnitt mit dem Namen des Druckers in der Konfigurationsdatei

hinterlegt und der Eintrag „**Active**“ gesetzt ist (=1, oder der erste Buchstabe ‚j‘ oder ‚y‘). Standardeintrag für „Active“ ist nicht gesetzt.

Bei nicht gesetztem „Active“, wird die Originalfunktion (winprint) aufgerufen. Ein Job-Logeintrag und u.U. auch Debuginträge werden trotzdem erzeugt.

Falls „**PrintSelf**“ gesetzt ist, wird die Ausgabe des Druckjob auf dem PrintMulti-Drucker auf dessen Port mit allen Einstellungen ausgegeben. Anderenfalls erfolgt keine Ausgabe. Dies ist auch der Standard. Die Originalausgabe funktioniert aber meistens nur dann, falls der ursprüngliche Drucker keinen eigenen Printprozessor, sondern „winprint“ verwendet. Die bei Windows mitgelieferten Treiber funktionieren meist.

Der Druckerabschnitt kann Aktionseinträge enthalten, die in der angegebenen Reihenfolge ausgeführt werden. Ein Aktionseintrag beginnt immer mit dem Wort „Action“. Der Rest ist beliebig. Auf der rechten Seite des Gleichheitszeichens steht der Type und der Abschnitt der Aktion und optional ein boolescher Wert (Standard *gesetzt*). Zu beachten ist, dass die Aktion nur ausgeführt wird, falls im Abschnitt der Wert von „Active“ auf gesetzt ist und der letzte Parameter nicht vorhanden oder „*gesetzt*“ ist. Der Standardwert für dieses „Active“ ist „*gesetzt*“ (anders als oben).

‘**Collate**’, ‘**Color**’, ‘**nUp**’, ‘**nUpBorder**’, ‘**DrvCopies**’, ‘**TotalCopies**’ wurden mit PrintMulti ab Version 1.0.2.1 eingeführt. Sie ermöglichen das Überschreiben der Einstellungen, welche der Benutzer beim Drucken eingestellt hat. Die Bedeutung der Einstellungen ist die gleiche wie bei der Print-Aktion später. Falls kein Wert konfiguriert wurde, wird die Originaleinstellung verwendet. Die Einstellungen wirken sich nur aus, falls auf den Originaldrucker physikalisch ausgegeben wird. Dies ist der Fall, falls “PrintSelf” gesetzt oder PrintMulti mit “Active=0” deaktiviert wurde.

```
[PrintMultiPrinter]
Active=1                               (Default:0)
ActionEgalWasHierSteht=Print;Section1  (wird ausgeführt)
Action2=Print;Section2                 (wird ausgeführt)
Action3=Print;Section3;1               (wird nicht ausgeführt)
Action4=Print;Section1;0               (wird nicht ausgeführt)

[Section1]
Active=1                               (Default: 1)
...
[Section2]
...
[Section3]
Active=0
```

Die Aktion „Print“

Folgende Einstellungen beeinflussen das Drucken auf den „Slave-Druckern“:

| Aktion Print | | | |
|------------------------|--------|------------------|---|
| Einstellung | Type | Default | Bedeutung |
| Active | Bool | 1 | Aktion benutzen ? |
| Printer | String | Name der Section | Name des Druckers, auf den gedruckt werden soll. |
| UseSystemAccount | Bool | 0 | Mit dem Systemaccount drucken (normalerweise nicht notwendig). |
| nUp | Int | *0) | Anzahl Seiten auf einer Druckseite. Mögliche Einstellungen: 1,2,4,6,9,16 |
| nUpBorder | Bool | *0) | Rahmen um die einzelnen Druckseiten |
| Booklet | Bool | *0) | Broschürendruck |
| Duplex | String | Simplex | „s“ oder „S“ für Simplex, „v“ oder „V“ für Duplex vertikal und „h“ bzw. „H“ für Duplex horizontal |
| Reverse | Bool | *0) | Letzte Seite zuerst drucken. Funktioniert mit einigen anderen Optionen nicht zusammen. |
| Collate | Bool | *0) | Sortieren (Seitenreihenfolge: 1-2-3-1-2-3 anstatt 1-1-2-2-3-3) |
| Color | Bool | Aus Devmode | Schwarzweißdruck für Farbdrucker erzwingen ? |
| DrvCopies | Int | *1) | Kopien für den Devicemode |
| TotalCopies | Int | *1) | Gesamtzahl von Kopien in PrintMulti |
| FirstPage | Int | 1 | Druckbereich |
| LastPage | Int | Anz. Seiten | |
| Devmodel | String | - | Device mode aus Datei |
| PaperSize | Int | | Papiergröße beim Zieldrucker |
| PaperSizeConversion | String | | |
| PaperSizeConvertAlways | Bool | | |
| PaperSource1 | Int | | Papierquellen für diverse Seiten |
| PaperSourceN | Int | | |
| PaperSourceL | Int | | |
| PaperSourceConversion | String | | |
| Save2File | String | „“ | Die Druckausgabe wird in die angegebene Datei gespeichert. Der Dateiname kann Macros enthalten. |
| Append2File | Bool | False | An vorhandene Datei anhängen. Nur für Save2File sinnvoll. |

| | | | |
|------------------------|---------------|--------------|--|
| Execute | Bool | False | Einstellungen für das Ausführen von Programmen auf die erzeugten RAW-Dateien. Siehe Kapitel „Execute-Einstellungen“ |
| ExecuteCmd | String | - | |
| ExecuteCurDir | String | - | |
| ExecuteAddPath | String | - | |
| ExecuteFlags | Int | 0 | |
| ExecuteTimeout | Int | n. warten | |
| ExecuteAsUser | Bool | False | |
| ExecuteShowWnd | Int | Normal | |
| WriteTextToFile | String | - | Schreibt den extrahieren Text aus dem gedruckten Dokument in die angegebene Datei |
| RefreshConfig | Bool | False | Liest die Konfigurationsdatei nach dem aktuellen Abschnitt erneut ein. Sinnvoll für dynamische Änderungen an der Konfigurationsdatei. |

- *0) Der Standard für diese Werte stammt aus den Jobattributen beim Drucken. Bei bestimmten Druckertreibern kann die Anzahl der Seiten und der Broschürendruck (nur bei Duplexdruckern) über die Druckereinstellungen vorgegeben werden.
- *1) Die Anzahl der Kopien sollten nicht immer vom Hauptdruckjob übernommen werden. Falls Sie z.B. PrintMulti zum gleichzeitigen Archivieren eines Dokuments verwenden wollen, sollten ja nicht mehrere Kopien eines gedruckten Dokuments im Archiv landen. Zu diesem Zweck können Sie die beiden Werte „DrvCopies“ und „TotalCopies“ setzen. Falls nicht festgelegt, dann enthalten beide Werte, die beim Drucken festgelegte Anzahl von Kopien. „TotalCopies“ ist die Gesamtzahl, „DrvCopies“ der Wert, der bei jedem Durchgang dem Slave-Drucker in seinen Device-Mode geschrieben und von der Totalzahl abgezogen wird.
Beispiel: TotalCopies 7, DrvCopies 3 ergibt drei Druckdurchläufe mit 3, 3 und 1 Kopie im Devicemode.

PaperSize-Einstellungen

Oftmals verfügen die Slave-Drucker nicht über die Papiergröße, die beim PrintMulti-Drucker für die Ausgabe eingestellt wurde (z.B. Din A3 bei A4-Druckern).

Durch Konfigurationseinstellungen muss deshalb PrintMulti beim Setzen des Zielformats auf die Sprünge geholfen werden.

Ein Setzen der Einstellung „PaperSize“ bewirkt, dass immer das angegebene Format verwendet wird unabhängig von der verwendeten Papiergröße beim Hauptdrucker.

Flexibler ist die Verwendung der Einstellung „PaperSizeConversion“. Es wird dort eine Abbildung definiert, welches Zielformat für ein Quellformat ersetzt werden soll.

Format ist Quellwert gefolgt von „->“ und dem Zielwert. Mehrere Abbildungen können durch Strichpunkt getrennt werden. Eine Quelle von „0“ gilt als Standardformat, welches verwendet wird, falls keine Abbildung definiert wurde.

Beispiel: „A3->Exe;0->A4;1->Letter;A4->B5“. Als Formate können die Papierformate oder die entsprechenden Windows-Konstanten verwendet werden (siehe Anhang).

Falls „PaperSizeConvertAlways“ wahr ist, wird die Abbildung immer angewendet. Im anderen Fall nur dann, falls das Quellformat beim Zieldrucker nicht verfügbar ist. Falls Werte nicht existieren, wird immer das Standardformat des Zieldruckers verwendet.

PaperSource-Einstellungen

Für die Auswahl der Papierschächte gilt ähnliches wie bei den Papierformaten. Falls Werte für die erste Seite, die Folgeseiten und/oder die letzte Seite angegeben sind, dann werden diese genutzt (bei einer Seite wird „PaperSource1“, bei zwei „PaperSource1“ und „PaperSourceL“ angewandt).

Einstellungen wirken auf Folgeseiten weiter (Falls nur „PaperSource1“ angegeben, dann gilt der Werte für alle Seiten).

Die Zahlenwerte und Namen der Papierschächte sind leider abhängig vom Druckertreiber. Das erschwert die Angabe der Abbildung. Namen werden immer im Kontext des Zieldruckers gesucht und zwar zunächst nach dem genauen Namen und falls nicht gefunden, nach übereinstimmendem Wortanfang.

Beispiel: „manu->Unter;3->2;15->auto“ (ersetzt z.B. „manuelle Auswahl“ oder „automatische Auswahl“)

Execute-Einstellungen

Falls „Execute“ gesetzt ist, wird nach dem Druckvorgang ein Prozess ausgeführt. Falls in eine Datei gespeichert wurde („Save2File“), dann wird das Makro „#G“ bzw. „#g“ mit diesem Dateinamen, ansonsten mit dem Namen der Spooldatei, verknüpft. Der auszuführende Prozess wird mit „ExecuteCmd“ angegeben. Der aktuelle Pfad kann mit „ExecuteCurDir“ eingestellt werden.

Falls „ExecuteTimeout“ > 0, wird die angegebene Zeit in ms auf das Ende des Prozesses gewartet. Bei „INF“ (= INFINITE) bis der Prozess zu Ende ist, bei „0“ wird gar nicht gewartet.

Der Wert von „ExecuteFlags“ und „ExecuteShowWnd“ kann der Beschreibung der Windows-API-Funktion „CreateProcess“ entnommen werden. Ein sinnvoller Wert für „ExecuteFlags“ kann 0x08000000 sein. Dann wird kein Konsolenfenster bei der Ausführung von Batchdateien angezeigt.

Falls „ExecuteAsUser“ falsch, dann wird der Prozess als „System“ ausgeführt. Im anderen Fall als der druckende Benutzer. „ExecuteShowWnd“ wird normalerweise 1 sein (SW_SHOWNORMAL).

Achtung!!! Für den Benutzer steht keine Registryzweig „HKEY_CURRENT_USER“ zur Verfügung. Registrydaten werden i.A. aus „HKEY_USERS\DEFAULT“ genutzt. Kopieren Sie ggf. Werte dorthin (u.U. für Netzwerk und Datenbankzugriffe notwendig).

Es werden einige Umgebungsvariablen bei der Ausführung des Programms gesetzt. Die Werte sind im Anhang aufgeführt. Zusätzlich werden die Variablen von „ExecuteAddPath“ an die „PATH“-Variable angehängt.

Der Abschnitt kann verwendet werden um mit Hilfe von Ghostscript PDF-Dateien zu erzeugen (siehe „PDF Erzeugung mit PrintMulti“).

Neu: Textextraktion

Ab Version 1.0.2.2 ist es möglich den Inhalt des gedruckten Dokuments für verschiedene Zwecke zu verwenden.

Das funktioniert aber nicht mit allen Anwendungen. Insbesondere beim Drucken von PDF Dateien ist es meist nicht möglich den Text zu extrahieren.

Zunächst steht nur der unformatierte Text zu Verfügung. Alle Textausgaben werden dazu direkt hintereinander in einen String gespeichert. Bei einem Seitenwechsel wird ein Zeilenumbruch erzeugt.

Es gibt zwei Möglichkeiten auf den Text zuzugreifen:

Zum einen kann der Text mit der Option „WriteTextToFile“ in einer Druckaktion in eine Datei gespeichert werden. Dies geschieht vor der Ausführung von Scripten mit „ExecuteCmd“, so dass innerhalb des Script die Datei gelesen und der Inhalt verwendet werden kann. Die zu verwendeten Textstellen sollten besonders gekennzeichnet sein (z.B.: eingeschlossen durch Sonderzeichen) und können auch sehr klein und mit weißer Schrift unsichtbar gedruckt werden.

Zum anderen gibt es ein das Makro „rawpage“ um innerhalb von Bedingungen auf den Seitentext zugreifen zu können. Mit einem Parameter können auch einzelne Seiten referenziert werden.

„rawpage“ ist insbesondere mit den neuen Operationen „tblsearch“ und „tbllookup“ sinnvoll, um in einer Tabelle (= Abschnitt in einer Konfigurationsdatei) Abbildungen zu ermöglichen. Es gibt (später) ein umfangreiches Beispiel zu diesem Thema.

Verwendung gespeicherter Druckereinstellungen (Schalter: Devmode1)

Als Parameter (der auch Bedingungen enthalten kann), wird der Name einer mit „Devmode2File“ erzeugten Datei erwartet. Falls die Treiberversionen und die Größe mit denen des gewählten Druckers übereinstimmen, werden die Einstellungen aus der Datei anstelle der Standardeinstellungen des Druckers verwendet. Der „Devicename“ (auf 30 Zeichen gekürzter Druckername) muss nicht unbedingt übereinstimmen. Dadurch

können die gespeicherten Druckereinstellungen (auch „device mode“) für verschiedene Drucker mit dem gleichen Treiber verwendet werden.

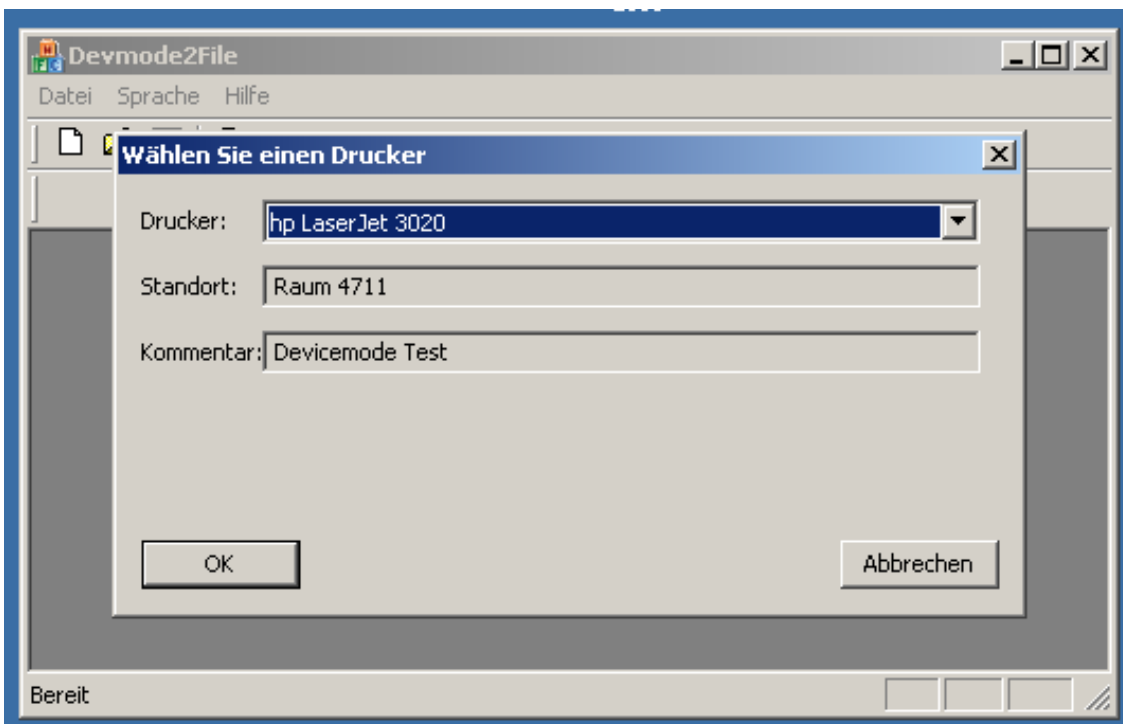
Die Möglichkeit kann genutzt werden um treiberspezifische Einstellungen wie z.B. Ausgabeschächte oder Wasserzeichen mit PrintMulti zu nutzen.

Die Druckereinstellungen gelten momentan **immer für alle** Seiten. Eine Änderung für andere Seitenbereiche wird bei Bedarf noch implementiert. Aus diesem Grund wurde der Name der Einstellung bereits wie bei den „Papersource“-Einstellungen gewählt.

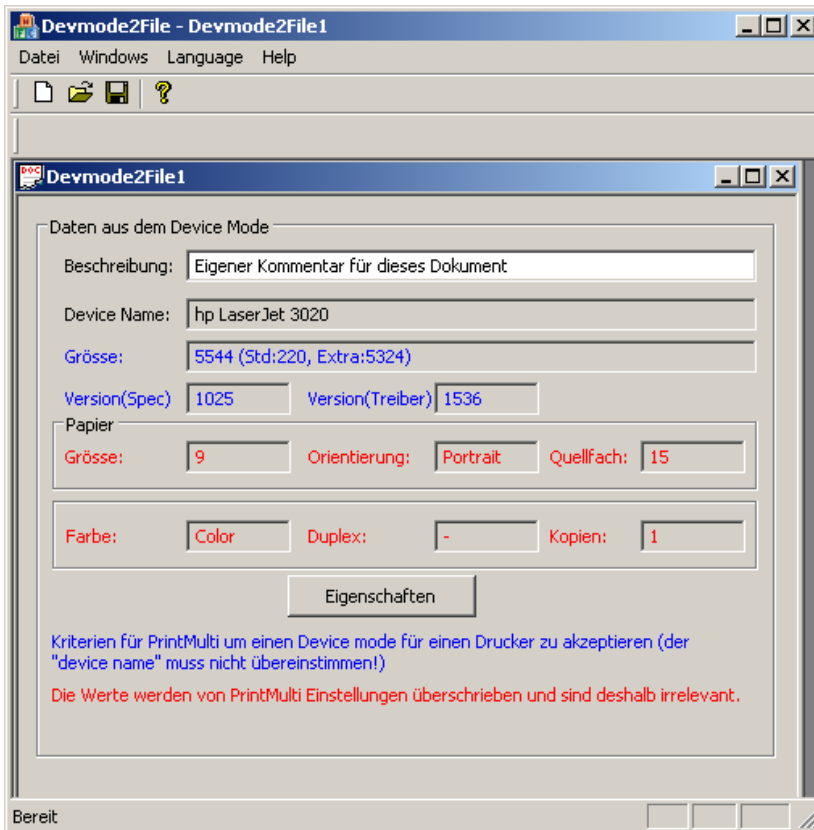
Ein Workaround ist das mehrmalige Ausgeben mit Einschränkungen des Druckbereichs über „FirstPage“ und „LastPage“.

Folgendes Beispiel soll die Verwendung von „Devmode2File“ und die Konfigurationseinstellungen erläutern: *„Einrichtung eines Druckers, der jeden Druckjob einmal normal und einmal mit einem Wasserzeichen „Kopie“ versehen ausgibt. Der Druckertreiber (hier HP Laserjet 3020 muss diese Möglichkeiten unterstützen.“*

Sie starten „Devmode2File“, wählen „Neu“ und selektieren den entsprechenden Drucker.



Nach Bestätigen mit OK erscheint folgender Dialog:

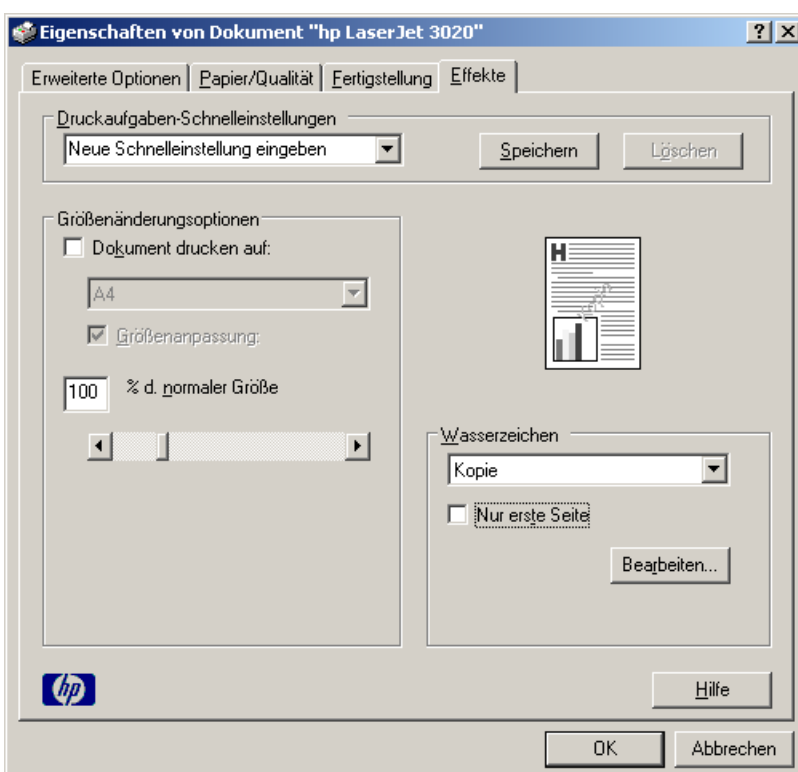


Die „Beschreibung“ ist das einzige Feld das geändert werden kann. Sie können hier eine Bezeichnung für die Konfiguration eingeben.

Die **blauen** Werte werden von PrintMulti bei der Auswahl eines Druckers überprüft. Sie müssen genau übereinstimmen damit die Daten verwendet werden.

Die **roten** Werte können bei den „Eigenschaften“ verändert werden. Sie werden aber durch andere PrintMulti Einstellungen überschrieben und sind deshalb irrelevant.

Die hier gewünschten Daten wie der Ausgabeschacht und das Wasserzeichen sind treiberspezifisch. Sie können nicht abgefragt und angezeigt werden. Einstellen können Sie die Daten aber nach Drücken des Buttons „Eigenschaften“. Der folgende Screenshot zeigt die Anpassung des Wasserzeichens. Andere Veränderungen können ebenso vorgenommen werden.



Anschließend können Sie die Datei unter einem vernünftigen Namen speichern („z.B. „C:\programme\PrintMulti\DeviceModi\HP3020_Kopie.dev“) und die PrintMulti.ini entsprechend anpassen:

```
[PrintWithCopy]
Active=1
ActionNormal=Print;ActionNormal
ActionKopie=Print;ActionKopie

[ActionNormal]
Printer=hp LaserJet 3020

[ActionKopie]
Printer=hp LaserJet 3020
Devmodel=%PM_INSTALLPATH%\DeviceModi\HP3020_Kopie.dev
```

Devmode2File verknüpft sich mit der Endung “*.dev”, so dass ein Doppelklick auf eine entsprechende Datei die Anwendung automatisch öffnet. Auch Drag & Drop ist möglich.

Achtung!

Der Name des Druckertreibers ist nicht im gespeicherten "Device Mode" enthalten. Beim Laden der Datei werden nur die Größen und die Versionen getestet (es ist nicht wünschenswert nur Dateien mit identischem Druckernamen zu verwenden). In den meisten Fällen wird der Test unpassende Dateien aussortieren. Die Benutzung einer Datei mit einem anderen Treiber kann zum Absturz des Spoolers führen. Ein Abfangen des Fehlers ist in PrintMulti nicht möglich (Exceptions in den Treibern).

Achten Sie bitte deshalb darauf nur passende Devicemode-Dateien zu verwenden.

Die Aktion „Exec“

Mit dieser Aktion kann ein Programm auf der Spooldatei selber ausgeführt werden. Mit Hilfe des SplViewer kann damit z.B. eine Druckvorschau realisiert werden. Aus dem Viewer kann der Druckjob dann wieder auf beliebigen Druckern mit diversen Optionen ausgegeben werden.

Falls die Aktion in einem eigenen T+hread ausgeführt werden soll, damit der Drucker so bald wie möglich wieder bereit ist, wird eine Kopie des Spooljobs erzeugt.

Es wird auf jeden Fall auf das Ende des Prozesses gewartet (Falls kein eigener Thread verwendet wird, ist der Drucker solange blockiert).

Hier die Einstellungen im Einzelnen:

| Aktion Exec | | | |
|---|------------------------------|--------------------------|--|
| Einstellung | Type | Default | Bedeutung |
| Active | Bool | 1 | Aktion benutzen ? |
| NewThread | Bool | False | Ausführung in eigenem Thread |
| ExecuteCmd ExecuteFlags ExecuteAsUser ExecuteShowWnd | String Int Bool Int | Siehe Aktion Print | Einstellungen für das Ausführen von Programmen auf die Spooldateien |
| RefreshConfig | Bool | False | Liest die Konfigurationsdatei nach dem aktuellen Abschnitt erneut ein. Sinnvoll für dynamische Änderungen an der Konfigurationsdatei. |

Beachten Sie, dass der SplViewer beim Öffnen der Spooldatei normalerweise keine Netzwerkdrucker kennt. Das liegt daran, dass die Verknüpfungen zu den Netzwerkdruckern in der Registry im Zweig HKEY_CURRENT_USER gespeichert sind, welcher nicht zur Verfügung steht.

Durch Kopieren der Einstellungen von

HKEY_CURRENT_USER\Printers

nach

KEY_USER\.Default\Printer

kann das Problem umgangen werden. Eine andere Möglichkeit besteht, wie immer bei Problemen mit Netzwerkdruckern, diese lokal anzulegen und nur den Port des Druckers oder der Ip-Adresse des Druckers zu verwenden.

Beispiel:

```
[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
```

Die Aktion „SaveEmf“

Die von PrintMulti verarbeitenden Spooldateien bestehen aus einzelnen Seiten im Emf Format. Mit dieser Aktion wird jede Seite in eine eigene Datei geschrieben und kann ggf. weiterverarbeitet werden. Die Aktion "SaveImage" ermöglicht ebenfalls ein Speichern im Emf-Format mit zusätzlicher Möglichkeit der Scriptausführung.

Hier die Einstellungen im Einzelnen:

| Aktion Exec | | | |
|-------------|--------|---------|----------------------------------|
| Einstellung | Type | Default | Bedeutung |
| Active | Bool | 1 | Aktion benutzen ? |
| Destination | String | | Pfad und Basisname der Emf-Datei |

Beispiel:

```
[ActionSaveEmf]
```

```
Active=1
```

```
Destination=c:\temp\#P_#J
```

erzeugt Dateien im Verzeichnis "c:\temp", die den Druckernamen, die JobId und die aktuelle Seite auf fünf Stellen formatiert enthalten.

Für einen Drucker "PrintMulti" und den ersten Job würden diese lauten:

```
PrintMulti_1_00001.emf
```

```
PrintMulti_1_00002.emf
```

```
...
```

PDF- und XPS Erzeugung mit PrintMulti

In diesem Kapitel werden drei Möglichkeiten erläutert, wie Sie automatisch PDF-Dateien mit PrintMulti erzeugen können. Je nach Aufgabenstellung wird die eine oder andere Lösung im bevorzugt werden.

Verwendung von zweistufigen PDF-Druckern

Zu dieser Art von PDF-Druckern gehören die meisten Freeware-Produkte (z.B: qvPDF, PDFCreator, FreePDF, eDocPrinter, Jaws PDF Creator aber auch Acrobat mit seinem PDF-Drucker/Destiller Gespann).

Alle Programme dieser Kategorie haben die Eigenschaft, dass die Anwendung auf einen Postscript-Drucker druckt und die Ausgabe durch ein weiteres Programm geschleust wird, welche die Postscriptdatei in PDF wandelt. „Umwandlungsprogramme“ sind „Ghostscript“, der „Acrobat Destiller“ oder eigene Tools.

Allen Tools ist gemeinsam, dass ein Drucken von Dokumenten in eine Datei zu einer Postscript und **nicht** zu einer PDF Datei führt. Das Verhalten ist auch mit anderen Programmen, wie Word zu beobachten. Es nützt also nichts, in PrintMulti die „Save2File“ Option zu benutzen, denn die Ausgabedatei enthält dann nur Postscript-Daten. Die Lösung ist ein Script mit „ExecuteCmd“ zu verwenden (sieht dritter Abschnitt „Direkte Erzeugung mit PrintMulti“) oder die Möglichkeiten der jeweiligen PDF Drucker zu verwenden. Der Name der PDF-Datei kann durch PrintMulti nicht beeinflusst werden.

Die Drucker bieten sich an, falls die oft weitgehenden Möglichkeiten zur anschließenden PDF-Weiterverarbeitung verwendet werden sollen (z.B: E-Mail Versand,... - qvPDF bietet hier sehr weitgehende Möglichkeiten).

Es folgt eine Beispielkonfiguration für den PDF-Drucker: „qvPDF“ und den physikalischer Drucker: „Laserjet“. Als PrintMulti Drucker dient „PDFLaserjet“. PDF-Ausgabeoptionen müssen innerhalb des qvPDF eingestellt werden.

```
[PDFLaserjet]
Active=1
PrintSelf=0
ActionPDF=Print;PDF
ActionPrint=Print;Laser

[PDF]
Active=1
Printer=qvPDF

[Laser]
Active=1
Printer=Laserjet
```

Verwendung von „nativen“ PDF/XPS -Druckern

Es gibt auch PDF Drucker, die direkt eine PDF Datei erstellen können. Das funktioniert dann sowohl mit PrintMulti als auch mit Word.

Wir können in dieser Kategorie den „Amyuni PDF Converter“ empfehlen (da wir ihn selber einsetzen). Bei der Fülle an PDF Lösungen gibt es aber sicher noch weitere Produkte, die gleichartig aufgebaut sind und reibungslos funktionieren.

Für die XPS Erzeugung kann der „Microsoft XPS Document Writer“ verwendet werden. Er ist ebenfalls ein „nativer“ Druckertreiber.

Der Name der Ausgabedatei kann durch die „Save2File“ Option eingestellt werden.

Hier ein Beispiel für die Erzeugung einer PDF und einer XPS Datei mit gleichzeitigem Ausdruck auf den Drucker „Laserjet“.

```
[PDFXPSLaserjet]
Active=1
PrintSelf=0
ActionPDF=Print;PDF
ActionXPS=Print;XPS
ActionPrint=Print;Laser

[PDF]
Active=1
Save2File=D:\PDFFiles\#P\#(%Y-%m)S\#K_#C.PDF
Printer=Amyuni PDF Converter

[XPS]
Active=1
Save2File=D:\XPSFiles\#P\#(%Y-%m)S\#K_#C.XPS
Printer=Microsoft XPS Document Writer

[Laser]
Active=1
Printer=Laserjet
```

Für die Ausgabe einer Testseite und der aktuellen ID 1 für den Drucker werden im Beispiel die beiden Dateien:

„D:\PDFFiles\PDFXPSLaserjet\2008-06\Testseite_1.pdf“ und

„D:\XPSFiles\PDFXPSLaserjet\2008-06\Testseite_1.xps“ erzeugt

Direkte Erzeugung mit PrintMulti (Ghostscript)

Sie benötigen hierfür ein installiertes Ghostscript. Falls Sie PDF/A kompatible Dokumente erzeugen wollen, sollte Sie Ghostscript ab Version 8.61 einsetzen. Tests mit dem sehr kritischen pdfaPiloten von Callas Software zeigen die volle PDF/A Kompatibilität (die wird von Acrobat 8.0 nicht immer erreicht!!!). Der Pilot kann übrigens auch zur Konvertierung von nicht PDF/A konformen Dokumenten in korrekte Dokumente verwendet werden.

Zusätzlich benötigen Sie einen geeigneten Postscript-Druckertreiber. Es kann unter anderen der bei Ghostscript mitgelieferte „Ghostscript PDF“ („ghostpdf.inf“ im lib-Unterverzeichnis) oder der „Rumborak PDF-Writer Plus“ (<http://www.rumborak.de/produktives>) verwendet werden.

Diese Treiber sind in der Lage Destiller Befehle in den Postscript-Datenstrom einzuschleusen, welche die spätere PDF-Erstellung mit Ghostscript beeinflussen.

Der Aufruf von Ghostscript wird durch die mitgelieferte VBScript-Datei „gscreatepdf1.vbs“ im Unterverzeichnis „PDFCreation“ durchgeführt. Die Datei soll als Muster für eigene Erweiterungen,

insbesondere für die Weiterverarbeitung der entstehenden PDF-Datei, dienen.

PrintMulti bietet einige Unterstützung für den Zugriff auf druckerrelevante Informationen innerhalb der Batchdatei, die durch die Einstellung „ExecuteCmd“ festgelegt wird.

1. Durch Kommandozeilenparameter, die Makros enthalten können (z.B. #G).
2. Durch Umgebungsvariablen die automatisch gesetzt werden (siehe Anhang, z.B. „PM_PRINTER“)
3. Durch Umgebungsvariablen, die im Druckerabschnitt der PrintMulti.ini festgelegt werden. Diese müssen mit einem Ausrufezeichen beginnen (z.B.: !GS_PDFMODE=PDF/A)

Das Beispiel wurde in der Version 1.0.3.0 um die Möglichkeit eines automatischen E-Mailversands erweitert. Weitere Informationen finden Sie im Kapitel mit den Beispielen.

Bitte beachten Sie, dass eine volle PDF/A Kompatibilität scheinbar nur erreicht wird, falls sie bei dem jeweiligen Postscriptdrucker folgende Einstellungen verwenden:

TrueType-Schriftart:"Als Softfont in den Drucker laden"

TrueType-Downloadoption:"Umriss"

Netzwerkprobleme

Ein PrintProcessor ist eine DLL, die vom Spoolerprozesses „spoolsv.exe“ geladen wird. Bei aktiver "Driver Isolation" (Windows 7 und Server 2008 R2) durch den Prozess "printisolationhost.exe". Der Spoolerprozess ist die ausführbare Datei des Druckerwarteschlangendienstes, der mit dem Systemkonto ausgeführt wird.

Das Systemkonto bietet weitreichende Administratorrechte, hat aber auch ein paar Einschränkungen:

1. Benutzt das Standardbenutzerprofil (HKEY_USERS\DEFAULT). Einstellungen von installierten Programmen müssen ggf. hierher kopiert werden. Das kann nötig sein beim Zugriff auf Datenbank oder auf Netzwerkdrucker.
2. Es ist nur ein Zugriff auf Netzwerkressourcen möglich, die Nullsessions erlauben. Wie ein Zugriff erlaubt werden kann, steht u.a. hier: <http://support.microsoft.com/kb/289655>

Kein Zugriff auf Netzwerkdrucker:

Falls der Zugriff auf Netzwerkdrucker nicht funktioniert, haben Sie drei Möglichkeiten:

1. Installieren der Drucker lokal. Als neuen „Local Port“ die Freigabe des Druckers auf dem Server mittels UNC-Namen angeben.
2. Kopieren der Daten von
“HKEY_CURRENT_USER\Printers\Connections“
“HKEY_CURRENT_USER\Printers\DevModePerUser“
“HKEY_CURRENT_USER\Printers\DevModes2“
nach
“HKEY_USERS\DEFAULT“
in den entsprechenden Unterzweig. Funktioniert aber nicht immer.
3. Experimentieren Sie mit der Einstellung „UseSystemAccount“ im Druckerabschnitt und versuchen den Nullsession-Zugriff für Druckerzugriffe zu erlauben.

Beispiele

Beispiele zu bedingten Ausdrücken

Seitenzahlabhängiges Drucken

Es könnte sinnvoll sein Druckjobs mit vielen Seiten auf einen leistungsfähigeren Drucker auszugeben. Unter der Annahme, dass der Drucker für die Benutzer „CommonPrinter“ heißt, der leistungsfähige „FastPrinter“ und der langsame „SlowPrinter“ würde sich bei einer Grenze von 100 Seiten folgende Konfigurationsdatei ergeben

```
[Common]
;32 für reine Calc-Meldungen. 59 für alle außer Job-Meldungen
LogMask=32
...
[CommonPrinter]
Active=1
ActionFast=Print;Fast;$(#Z;100;>=)
ActionSlow=Print;Slow;$(#Z;100;<)

[Fast]
Printer=FastPrinter

[Slow]
Printer=SlowPrinter
```

Hier noch eine andere Möglichkeit

```
[CommonPrinter]
Active=1
ActionSlowFast=Print;SlowFast

[SlowFast]
Printer=$(#Z;100;>=;FastPrinter;SlowPrinter;?)
```

Ein Ausschnitt aus der Logdatei sieht dann folgendermaßen aus:

```
CALC;...;print ;SlowFast;SlowFast.Printer: 1 >= 100 ->0
CALC;...;print ;SlowFast;SlowFast.Printer: (0)? FastPrinter : SlowPrinter ->SlowPrinter
```

Die Berechnungsreihenfolge ist hier gut erkennbar.

Benutzerabhängiges Drucken

Ein bestimmter Personenkreis könnte besondere Rechte haben. Dafür eignet sich der „contains“-Operator. Im Beispiel sollen bestimmte Benutzer farbig Drucken dürfen, alle anderen nicht. Eine einfachere Alternative mit dem „tbllookup“-Operator wird weiter unten beschrieben. Der Schwerpunkt hier liegt darin, den „contains“-Operator zu erläutern.

```
[Common]
_ColorUsers=|dieter|gustav|bärbel|
_IsColorUser=$(#(_ColorUsers)I;upper;|#U;+;|;+;upper;contains)

[CommonPrinter]
Active=1
ActionColor=Print;Color

[Color]
Printer=ColorPrinter
Color=$(#(Common._IsColorUser)I;1;0;?)
```

Ein Ausschnitt aus der Logdatei verdeutlicht den komplizierten Ausdruck „_IsColorUser“

```
CALC;...;Read entry #(Common._IsColorUser)I
CALC;...;Read entry #(Common._ColorUsers)I
CALC;...;Common._IsColorUser: Upper("|dieter|gustav|bärbel|")->
"|DIETER|GUSTAV|BÄRBEL|" Locale:German
CALC;...;Common._IsColorUser: "|" + "dieter" ->"|dieter"
CALC;...;Common._IsColorUser: "|dieter" + "|" ->"|dieter|"
CALC;...;Common._IsColorUser: Upper("|dieter|")->"|DIETER|" Locale:German
CALC;...;Common._IsColorUser: "|DIETER|GUSTAV|BÄRBEL|" contains "|DIETER|" ->1
CALC;...;Color.Color: (1)? 1 : 0 ->1
```

Folgende Punkte soll das Beispiel zeigen:

Benutzung des Makros # (...) I zur Verwendung eines anderen Abschnittes. Der Unterstrich vor dem Bezeichner ist nur wegen der Übersichtlichkeit (im Gegensatz zu einem führenden „!“ mit dem Umgebungsvariablen bei der Ausführung von Scripten gesetzt werden können).

Wie ein Textvergleich unabhängig von der Schreibweise durchgeführt werden kann. Hier wird im Gegensatz zu den Vergleichsoperationen (z.B.: „<“) keine Regionseinstellungen von Windows verwendet (hier „German“, siehe Logdatei).

Dass Ausdrücke immer neu berechnet werden und dadurch der aktuelle Kontext verwendet werden kann.

Es könnte in der letzten Zeile auch „Color=#(Common._IsColorUser)I,“ geschrieben werden, da Color nur die Werte 0 oder 1 akzeptiert. Die angegebene Zuweisung mit dem Conditional-Operator kann aber leicht auf andere Einstellungen übertragen werden.

Dass die Logdatei wichtig ist um komplizierte Ausdrücke zu verstehen.

Dokumentabhängiges Drucken

Sie wollen z.B.: anhand des Dokumentennamens ein bestimmtes Archiv setzen, das Sie in der Scriptdatei ansprechen wollen. Das Beispiel verwendet den Amyuni PDF Converter zur Erzeugung der PDF. Das Script sollte die Datei nach dem Importieren wieder löschen. Hier werden lediglich die übergebenen Parameter angezeigt.

```
[CommonPrinter]
Active=1
ActionArchiv=Print;Archiv

[Archiv]
Printer=Amyuni PDF Converter
Save2File=%TEMP%\#P_#C.PDF
_IsInvoice=$(#D;upper;INVOICE;contains)
_IsDelivery=$(#D;upper;DELIVERY;contains)
!ArchivName=$(#(_IsInvoice)I;INVOICE;#(_IsDelivery)I;DELIVERY;
UNKNOWN;???)
Execute=y
ExecuteCmd=cmd /K echo #G
```

```
CALC;...;Read entry #(Archiv._IsInvoice)I
CALC;...;Archiv._IsInvoice: Upper("Microsoft Word - MyInvoice.doc")->"MICROSOFT WORD -
MYINVOICE.DOC" Locale:German
CALC;...;Archiv._IsInvoice: "MICROSOFT WORD - MYINVOICE.DOC" contains "INVOICE" ->1
CALC;...;Read entry #(Archiv._IsDelivery)I
CALC;...;Archiv._IsDelivery: Upper("Microsoft Word - MyInvoice.doc")->"MICROSOFT WORD -
MYINVOICE.DOC" Locale:German
CALC;...;Archiv._IsDelivery: "MICROSOFT WORD - MYINVOICE.DOC" contains "DELIVERY" ->0
CALC;...;Archiv.!ArchivName: (0)? DELIVERY : UNKNOWN ->UNKNOWN
```


Das Beispiel öffnet ein Kommandozeilenfenster. Zur Erinnerung: Alle Variablen, die mit „!“ in einem Abschnitt vorkommen, werden als Umgebungsvariablen dem Befehl mitgegeben (ohne das Ausrufezeichen). Sie können den Archivnamen auch als Kommandozeilenparameter an das ausführbare Programm übergeben. Das würde dann so aussehen: „... #G #(!ArchivName) I“.

```

C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32>set
ArchivName=INVOICE

PM_ACTION=Archiv
PM_ACTTIME=2008-07-19 18:04:21
PM_COLOR=1
PM_COUNTER=261
PM_DUPLEX=1
PM_FILE=d:\temp\CommonPrinter_261.PDF
PM_INSTALLPATH=C:\Programme\PrintMulti
PM_JOBID=60
PM_MACHINE=
PM_PAGES=1
PM_PAPERSIZE=9
PM_PRINTER=CommonPrinter
PM_PRINTTIME=2008-07-19 18:04:20
PM_USER=

```

Einige interne Dinge wurden ausgegraut. Die erzeugten Umgebungsvariablen sind aber sehr gut zu erkennen.

Das Beispiel könnte auch etwas einfacher folgendermaßen konfiguriert werden. Falls keine der beiden Bedingungen zutrifft, wird allerdings hier nichts ausgegeben.

```

[CommonPrinter]
Active=1
ActionArchiv1=Print;InvoiceArchiv;$(#D;upper;INVOICE;contains)
ActionArchiv2=Print;DeliveryArchiv;$(#D;upper;DELIVERY;contains)

[InvoiceArchiv]
Printer=Amyuni PDF Converter
Save2File=%TEMP%\#P_#C.PDF
!ArchivName=INVOICE
Execute=y
ExecuteCmd=cmd /K echo #G

[DeliveryArchiv]
Printer=Amyuni PDF Converter
Save2File=%TEMP%\#P_#C.PDF
!ArchivName=DELIVERY
Execute=y
ExecuteCmd=cmd /K echo #G

```

Gleichbedeutend wäre die Bedingung hinter dem Action-Eintrag in das Flag „Active=“ innerhalb der einzelnen Sektionen zu verschieben.

```

[CommonPrinter]
Active=1
ActionArchiv1=Print;InvoiceArchiv
ActionArchiv2=Print;DeliveryArchiv

[InvoiceArchiv]
Active=$(#D;upper;INVOICE;contains)
...

```

Datumsabhängiges Drucken

Sie wollen z.B. zwischen acht Uhr abends und sechs Uhr morgens, warum auch immer, einen anderen Drucker verwenden.

```

[CommonPrinter]
Active=1
ActionNight=Print;NightPrint;$(#(%H)S;20;>=#(%H)S;6;<=;or)
ActionDay=Print;DayPrint;$(#(%H)S;20;<#(%H)S;6;<;and)

[NightPrint]
Printer=SilentPrinter

[DayPrint]
Printer=LoudPrinter

```

Die Logdatei würde beim Drucken zwischen 21 und 22 Uhr unter anderem folgende Einträge enthalten.

```

CALC      ;...;CommonPrinter.ActionNight: 21 >= 20 ->1
CALC      ;...;CommonPrinter.ActionNight: 21 <= 6 ->0
CALC      ;...;CommonPrinter.ActionNight: 1 OR 0 ->1
DEBUG     ;...;print ;NightPrint           ;Start Action:'NightPrint'

```

Nutzen des Seitentextes

Mit den neuen Befehlen bzw. Einstellungen „tblsearch“, „tbllookup“, „rawpage“, „writetexttofile“ ergeben sich neue Möglichkeiten Informationen aus dem gedruckten Text zu verwenden.

Der „tbllookup“- Operator ermöglicht ein einfaches Nachschlagen eines Begriffes in einem Konfigurationsabschnitt. Damit lässt sich das Beispiel zum „Benutzerabhängigen Drucker“ viel einfacher gestalten.

Der „tblsearch“ – Operator geht alle Einträge eines Abschnitts der Reihe nach durch und gibt den (berechneten) Wert auf der rechten Seite zurück, falls der Eintrag in dem entsprechenden Parameter (unten „1;rawpage“) enthalten ist.

Achtung!!! „tbllookup“ unterscheidet **nicht** zwischen Groß- und Kleinbuchstaben; „tblsearch“ schon.

```
[ColorUsers]
Dieter=$(#Z;100;<)
Gustav=0

[Printers]
#invoice#=PrinterInvoice
#delivery#=PrinterDelivery

[ActionTblTest]
Active=1
Color=$(;ColorUsers;#U;0;tbllookup)
Printer=$(;Printers;1;rawpage;TestPrinter;tblsearch)
; Example for a different configuration file and the use of all pages
; Printer=$(%PM_INSTALLPATH%\users.ini;;rawpage;TestPrinter;tblsearch)
_TextFile=%TEMP%\#P_#C.txt
WriteTextToFile=#(_TextFile)I
Execute=y
ExecuteCmd=wscript "%PM_INSTALLPATH%\PDFCreation\textextract.vbs" "#(_TextFile)I"
```

Hier noch ein paar Erläuterungen:

Auf der rechten Seite einer Zuweisung kann wieder ein auswertbarer Ausdruck stehen. „Dieter“ würde nur in Farbe drucken dürfen, bei Dokumenten mit weniger als 100 Seiten.

Falls die Einstellungen für „tblsearch“ oder „tbllookup“ auf eine andere Datei verweisen, wird diese bei jeder Auswertung neu geladen. Das erleichtert womöglich eine dynamische Änderung an den Konfigurationsdateien.

Der neue „rawpage“ Operator liefert einen Text, der alle gedruckten Texte hintereinander ohne Trennzeichen enthält. Als erster Parameter kann eine Seitennummer, beginnend bei eins angegeben werden. Ohne Angabe enthält der Text alle Seiten. Neue Seiten werden durch ein Zeilenende gekennzeichnet. Durch die fehlenden Trennzeichen müssen gesuchte Textpassagen durch besondere Zeichen eingefasst werden (hier ‚#‘). Der gedruckte Text kann unsichtbar sein durch die Verwendung von weißer Farbe.

„WriteTextToFile“ ist sinnvoll bei der Suche nach Problemen, falls etwas nicht gefunden werden sollte. Die Datei kann auch wie im Beispiel durch ein Script ausgewertet werden. Das Script „textextract.vbs“ wird mitgeliefert. Das Format der Textdatei ist UTF16.

Spezielles Protokollieren von Jobs

Nehmen wir an, Sie wollen in einer Datenbank für jeden Druckjob vermerken, welcher Benutzer auf welchen Drucker wie viele Seiten und Kopien in Farbe bzw. Schwarzweiß gedruckt hat. Sie können natürlich auch die Logdateien untersuchen. In diesem Beispiel wird aber eine Programmausführung dazu verwendet:

Sie richten dazu einen Drucker „SumPages“ ein und weisen ihm den PrintProzessor „PrintMulti“ zu.

Der Ini-Abschnitt sieht folgendermassen aus:

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

„sumpages.bat“ würde den Code enthalten, der die übergebenen Daten in eine Datenbanktabelle einfügt. Hier wurde nur eine Batchdatei verwendet, welche die Parameter an eine CSV-Datei anhängt:

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

In der Zieldatei wird jetzt für jeden Druckjob eine Zeile erzeugt mit dem Benutzer, dem Drucker, 1 bei Farbe / 0 SW, Anzahl der Seiten und Anzahl der Kopien.

Hier noch ein paar Erläuterungen:

Die Aktion weist auf sich selber - dient also als Haupteintrag und „Actioneintrag“. Ein direkter rekursiver Aufruf von PrintMulti wird intern unterdrückt. Derartige Versuche über mehrere „PrintMulti“-aktiven Drucker könnte zu einer Endlosschleife führen, also Vorsicht! Da kein „Printer“-Eintrag vorhanden ist, wird der Name des Abschnitts „SumPages“ als Druckername verwendet.

Nach dem Drucken wird das entsprechende Kommando ausgeführt. Die Flags bewirken, dass kein Konsolenfenster aufklappt.

Speichern von Druckjobs

Sie wollen z.B. alle Druckausgaben mitprotokollieren, so dass Sie später nochmals ausgedruckt oder an andere zum Drucken weitergegeben werden.

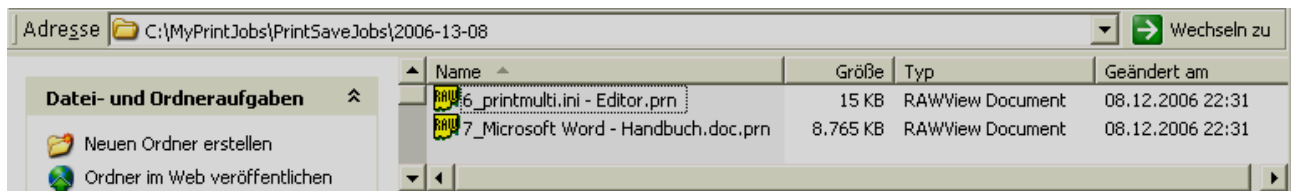
```
[PrintSaveJobs]
Active=1
Action1=Print;PrintJobs
Action2=Print;SaveJobs

;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
```

So würden die Dateien abgelegt:



Die Batchdatei „savepages.bat“ enthält nur eine Zeile:

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

Die Jobdatei würde nach den zwei Druckjobs folgendes enthalten:

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

Das Beispiel zeigt die Nutzung der Umgebungsvariablen, die im Anhang erläutert sind.

Übrigens kann „#D“ im Dateinamen zu Problemen führen. Falls Sie zum Beispiel aus einem Editor eine Datei drucken, enthält der Dokumentenname, je nach Editor, den vollständigen Pfad der bearbeiteten Datei, der beim Erzeugen der „Save2File“-Datei zu einem Fehler führen kann. Sie sollten anstatt „#D“ in diesem Fall „#K“ verwenden.

Es wird jetzt jeden Tag ein neues Verzeichnis angelegt und alle Druckjobs, die auf „PrintSaveJobs“ gedruckt werden dort abgelegt. Die CSV-Datei enthält Verweise und Informationen über die Datei und kann entsprechend ausgewertet werden.

Falls „Append2File“ auf 1 gesetzt wäre, dann würden die Daten an die Datei angehängt werden (dann sollte aber ein anderer Name gewählt werden). Das kann bei der Weitergabe an Dritte zum Drucken sinnvoll sein.

„#C“ ist ein eindeutiger Jobzähler für den Drucker „PrintSaveJobs“ im Beispiel, der bei jedem

Druckjob hochgezählt wird. Er wird gespeichert in:

```
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Printers\  
PrintSaveJobs\PrinterDriverData\JobCounter"
```

und kann dort ggf. zurückgesetzt werden.

Verarbeitung von PDF Dateien

Im Unterverzeichnis "PDFCreation" der PrintMulti Installation findet sich das Script "gscreatepdf1.vbs". Es wurde um die Möglichkeit erweitert PDF Dateien als Mailanhang an eine im Dokument stehende Adresse zu verschicken.

Die Grundaufgabe des Scriptes ist es PDF-Dateien (auch PDF/A) mit Hilfe von Ghostscript (=GS) zu erzeugen. Eine vorhandene GS Installation ist nötig und sollte automatisch erkannt werden.

Viele Eigenschaften werden durch feste Einträge in der PrintMulti.ini gesteuert. Sie könnten aber auch, wie einige andere Eigenschaften, über den gedruckten Text ausgewertet werden.

Zunächst einige Punkte über die PDF-Erzeugung:

```
[ActionCreatePDFScript]  
Printer=Ghostscript PDF  
Save2File=%TEMP%\#P_#C.ps  
Color=1  
Execute=yes  
ExecuteFlags=0x08000000  
ExecuteCmd=cscript "%PM_INSTALLPATH%\PDFCreation\gscreatepdf1.vbs" "#G"  
    "%TEMP%\#K_#C_#U.pdf"  
!GS_COMPATIBILITY=1.4  
; screen,ebook,printer,prepress,default  
!GS_PDFSETTINGS=screen  
; a0,a1,..,a10,c0,..,c6,11x17,ledger,legal,letter,... uncomment for fixed paper  
size  
;!GS_FIXPAPERSIZE=a0  
; Additional gswin32 parameter. See use.htm in gs\doc-path  
; Can also be appended to the command line below.  
;!GS_EXTRAPARAMS=-dPDFWRDEBUG  
; Append log messages to that file. Comment out for no debug messages  
!GS_LOGFILE=%TEMP%\out.txt  
; Set some internal parameters for PDF/A compatibility and use PDF_A_def.ps file  
!GS_PDFMODE=PDF/A
```

Sie benötigen zum Drucken einen Postscript Drucker. Der angegebene "Ghostscript PDF" ist in der GS Installation enthalten (inf-Datei im lib-Verzeichnis).

Die Postscriptdatei wird im Tempverzeichnis unter einem eindeutigen Namen gespeichert (#C ist ein Counter). Das ist normalerweise das "C:\Windows\Temp"-Verzeichnis, da das Script im Kontext des System-Account ausgeführt wird und keinen User-Registryzweig kennt.

Das Script bekommt die Postscriptdatei (#G) und die Zieldatei als Parameter. Für eine Debugausgabe müssen Sie "ExecuteFlags" auf 0 setzen und das Script mit "ExecuteCmd=cmd /k cscript ..." aufrufen.

Alle Parameter die mit einem Ausrufezeichen anfangen werden dem ausgeführten Programm als Environmentvariable übergeben.

Falls Sie !GS_LOGFILE setzen, enthält die Datei Ausgaben vom Aufruf von GS. Das ist für eine Fehlersuche wichtig. Nicht immer kann GS die Postscriptdateien erfolgreich wandeln.

Eine PDF/A Ausgabe verlangt ein Farbprofil. Bei der Installation von PrintMulti wird eines im entsprechenden Ordner von Windows gespeichert (siehe Installationshinweise am Anfang des Dokuments).

Hier ein Ausschnitt aus dem Script, das zeigt wie aus den Parametern eine Kommandozeile für den Aufruf von GS gebastelt wird:

```
compatibility = procEnv("GS_COMPATIBILITY")
pdfsetting    = procEnv("GS_PDFSETTINGS")
fixpapersize  = procEnv("GS_FIXPAPERSIZE")
debugoutput   = procEnv("GS_LOGFILE")
extraparams  = procEnv("GS_EXTRAPARAMS")
pdfmode       = procEnv("GS_PDFMODE")

'Build command line file for ghostscript
paraFileName=createGuid()+".txt"
Set fso = CreateObject("Scripting.FileSystemObject")

'Base cmd line
cmdLine = "-q -dSAFER -dNOPAUSE -dBATCH -sDEVICE=pdfwrite "

'PDFMODE has priority over some other settings
if ucase(pdfmode) = "PDF/A" Then
    pdfaxFile = createGuid()+".ps"
    ReplaceEnvironment "%PM_INSTALLPATH%\PDFCreation\PDFA_def.ps", pdfaxFile
    cmdLine = cmdLine + "-dPDFA -dUseCIEColor -sProcessColorModel=DeviceCMYK "
Else
    if compatibility <> "" Then cmdLine = cmdLine + "-
        CompatibilityLevel="+compatibility+" "
    if pdfsetting <> "" Then cmdLine = cmdLine + "-dPDFSetting=/" + pdfsetting + " "
    if fixpapersize <> "" Then cmdLine = cmdLine + "-sPAPERSIZE=" + fixpapersize + " "
    if extraparams <> "" Then cmdLine = cmdLine + extraparams + " "
End if
```

Automatischer Mailversand:

Der Mailversand wird im Script mit dem leistungsfähigen Tool "Blat.exe" (www.blat.net). Es gibt aber sicher auch andere Tools für den Mailversand per Kommandozeile, die es erlauben einen Anhang mitzuschicken.

```
; to test mail sending, uncomment the three following lines
!GS_SENDMAIL=1
_DstFile=%TEMP%\#P_#C.txt
WriteTextToFile=$((!GS_SENDMAIL)I;1;=#(_DstFile)I;?)

; Options used to send the printed file as attached pdf file to an address
; defined in the example document
!GS_MAILFILE=%PM_INSTALLPATH%\PDFCreation\mailbody.txt
; activate for a log file which shows the output of blat, which is used for
; mail sending
!GS_MAILLOGFILE=%TEMP%\mail.log

;server and login data
!GS_MAILSERVER=xxxxxxx
!GS_MAILSERVERUSER=xxxxxxx
!GS_MAILSERVERPASSWORD=xxxxxxx
!GS_MAILREPLYADDRESS=info@lvbprint.de

; set the name to be used in the script
!GS_TEXTFILE=#(_DstFile)I
```

Die Server und Logindaten sind in der PrintMulti.ini konfiguriert.

Zieladresse und Betreff werden im Word-Dokument übergeben (siehe SendMailExample.doc).

Der Nachrichtentext wird im Beispiel aus der festen Datei "mailbody.txt" genommen.

Falls "!GS_MAILLOGFILE" definiert ist, wird eine Logdatei geschrieben.

Der Schalter "!GS_SENDBMAIL" entscheidet, ob überhaupt eine Mail geschickt werden soll. Er wird auch im Script ausgewertet.

Falls der Schalter gesetzt ist, dann wird mit Hilfe des Befehls "WriteTextToFile" der unformatierte Seitentext in eine Datei im Tempverzeichnis geschrieben.

Falls nicht, wird der Ausdruck zu einem Leerstring ausgewertet (zu sehen in der Logdatei) und in diesem Fall ignoriert.

Der Name der Textdatei wird an die Umgebungsvariable "!GS_TEXTFILE" übergeben.

Die Textdatei wird übrigens im Script wieder gelöscht, damit keine überflüssigen Dateien zurück bleiben.

Die Extraktion der Daten aus dem Quelltext wird im Script mit der Funktion "ExtractDataFromText" durchgeführt.

Hier ein Ausschnitt aus dem Script.

```
' must be set in the printmulti.ini
sendMail = procEnv("GS_SENDBMAIL")

' needs blat.exe in the PDFCreation folder
' needs CmdRedirect.exe in the PDFCreation folder (set above)
' adjust log file, user name and password
If sendMail = "1" Then
    Dim blatExeFile, mailfile, subject, address, tempMailLogFile, mailLogFile
    Dim mailserver, mailserveruser, mailserverpassword, mailreplyaddress, strContent

    ' get the name of the text file
    textFileName = procEnv("GS_TEXTFILE")

    ' load the complete text into a variable
    set textFile = fso.OpenTextFile(textFileName, ForReading, false, TriStateTrue)
    strContent=textFile.ReadAll()
    textFile.Close

    ' get the address and the subject from the printed text
    address = ExtractDataFromText(strContent, "GS_MAILSENDERADDRESS")
    subject = ExtractDataFromText(strContent, "GS_MAILSUBJECT")

    blatExeFile=shell.ExpandEnvironmentStrings("%PM_INSTALLPATH%\PDFCreation\blat.exe")
    ' the rest is defined as constants in the PrintMulti.ini file
    mailfile = procEnv("GS_MAILFILE")
    mailLogFile = procEnv("GS_MAILLOGFILE")
    mailserver = procEnv("GS_MAILSERVER")
    mailserveruser = procEnv("GS_MAILSERVERUSER")
    mailserverpassword = procEnv("GS_MAILSERVERPASSWORD")
    mailreplyaddress = procEnv("GS_MAILREPLYADDRESS")

    gsCmdLine = ...

    shell.Run(gsCmdLine),0,true

    If mailLogFile <> "" Then
        AppendFile2File mailLogFile, tempMailLogFile, 2
        Wscript.echo mailLogFile + ":Append:"+tempMailLogFile
        fso.DeleteFile tempMailLogFile
    End If

    fso.DeleteFile textFileName
End if
```


Anhang

Umgebungsvariable bei der Ausführung von Programmen

| | |
|----------------|---|
| PM ACTION | Name der Aktion (#A) |
| PM COUNTER | Eindeutiger Zähler (#C) |
| PM JOBID | JobId von Windows (#J) |
| PM USER | Benutzername (#U) |
| PM MACHINE | Computername des druckenden Computers (#M) |
| PM PRINTER | Name des Hauptdruckers (#P) |
| PM PAGES | Anzahl von Seiten (#Z) |
| PM COLOR | Farbdruck (#c) |
| PM PAPERSIZE | Papiergröße (#s) |
| PM DUPLEX | Simples, Duplex Horizontal oder Simplex Vertikal (#d) |
| PM ACTTIME | Aktuelle Zeit (#T) |
| PM PRINTTIME | Zeit aus dem Druckjob (#S) |
| PM FILE | Druckausgabedatei (#G) |
| PM_INSTALLPATH | Installationspfad aus der Registry oder %Programfiles%\PrintMulti, falls nicht gesetzt. |

Papierformate

Folgende Papierformate werden als Text erkannt und den Werten in der ersten Spalte zugeordnet

| 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 |

Werte für „ExecuteShowWnd“

Die Zahlen sind zu verwenden

| | |
|--------------------|----|
| 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 |