

Technische Dokumentation zur Ansteuerung eines VIVA Network-Publishing Servers für Windows Version 4.0 (ab Version 9.5).- Stand Montag, 29. Mai 2017

DEFINITION	1	PARAMETER FÜR DIE AUSGABE	8
EINLEITUNG	1	print_printLayer (optional ab Version 9.1).....	8
PARAMETERANWENDUNG	2	print_imageQuality (optional).....	9
PARAMETERBESCHREIBUNG	2	Marken (optional).....	9
ALLGEMEINE PARAMETER	2	Anschnitt (optional).....	9
sender.....	3	PDF Optimierungen (optional ab Version 9.5).....	9
sourceType.....	3	PDF Optionen(optional).....	10
dataFile	4	ICC-Profile (optional).....	11
dataPath.....	4	Parameter für JPG, PNG, Bildausgabe (optional).....	11
templateName.....	4	Diverse Parameter (optional).....	12
templateFile.....	4	colorSeparation (optional).....	12
exportName.....	4	ZIP Parameter (optional).....	13
outType.....	4	ANWENDUNGSBEISPIELE	14
outHow.....	5	Ansteuerung über eine Programmiersprache.....	14
outRedirectionUrl (optional).....	5	Ansteuerung über eine Datenbank.....	14
waitType (optional).....	6	Ansteuerung über einen Internet-Browser.....	14
waitTimeout (optional).....	6	ANSTEUERUNG ÜBER DIE DOS-KOMMANDOZEILE	15
dstPath (optional).....	6	GEÄNDERTE VARIABLENNAMEN	16
outDevice (optional).....	6	GEÄNDERTE VARIABLENANWENDUNG	16
pagesSource (optional).....	7	OBSOLETE FUNKTIONEN	16
firstPage (optional).....	7	outRedirectionUrl (optional).....	16
lastPage (optional).....	7		
language (optional).....	8		
label (optional).....	8		
ticketID (optional).....	8		

Copyright

© 1994-2017 by VIVA. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage and the VIVA copyright notice appears. If the majority of the document is copied or redistributed, it must be distributed verbatim, without repagination or reformatting. To copy otherwise requires specific permission from VIVA.

Licenses and Trademarks

VIVA, VivaDesigner, VivaImpose, VivaPosterize, VivaXML, VIVA Network-Publishing, VivaIP, VivaCP, Intelligent Publishing, Corporate Publishing, Enterprise Publishing, Database Publishing, VIVA Creative Publishing are trademarks of VIVA or its subsidiaries, and may be registered in some jurisdictions.

PostScript and PDF are trademarks of Adobe Systems Incorporated. All instances of the name PostScript and PDF in the text are references to the PostScript and PDF language as defined by Adobe Systems Incorporated unless otherwise stated. PostScript, PDF, TIFF and Adobe are trademarks of Adobe Systems Incorporated or its subsidiaries, and may be registered in some jurisdictions.

Apple, and Macintosh are registered trademarks are trademarks of Apple, Computer, Inc. Microsoft and MS-DOS are registered trademarks and Windows is a trademark of Microsoft Corporation.

UNIX is a registered trademark of UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc.

All other trademarks are the property of their respective owners. E & OE.

Production Notes

This document was created electronically using VivaDesigner.

DEFINITION

Diese Dokumentation beschreibt die Kommunikation zwischen einer Anwendung und dem VIVA Network-Publishing Server. Sie beschreibt nicht die grundsätzliche Arbeitsweise des VIVA Network-Publishing Servers, die in einer anderen Dokumentation erläutert wird.

In dieser Dokumentation werden folgende Begriffe verwendet:

1. **Anfrage**
Als Anfrage bezeichnen wir die Übertragung der Daten und der Parameter an den »VIVA Network-Publishing Server«, mit dem Auftrag, entsprechende Dokumente zu erstellen.
2. **Daten**
Als Daten bezeichnen wir die Informationen, die vom »VIVA Network-Publishing Server« verarbeitet werden sollen (Datendatei, Parameter, etc.).
3. **Parameter**
Als Parameter bezeichnen wir die Informationen, die dem VIVA Network-Publishing Server mitteilen, in welchem Format und mit welchen Einstellungen (Wert) die Ergebnisdatei erstellt werden soll.
4. **Datendatei**
Als Datendatei bezeichnen wir die Textdatei (.csv-Format), die die darzustellenden Informationen enthält. Bilder werden in Form von Namen und ggf. Verzeichnissen repräsentiert. Die Datendatei kann bei großen Datenmengen auch komprimiert (.gz) übertragen werden.
5. **Ergebnisdatei**
Als Ergebnisdatei bezeichnen wir das Dokument, das der VIVA Network-Publishing Server aufgrund der Daten und Parameter erstellt.
6. **Template**
Eine Vorlage, die alle Informationen und Regelwerke enthält, wie die Daten verarbeitet und grafisch umgesetzt werden sollen. Ein Template ist eine native VivaNWP-Designer-Datei.
7. **VivaNWP-Server (VIVA Network Publishing Server)**
Ein »VivaNWP-Server« ist ein Publishing Server, auf dem neben der VivaNWP-Publishing-Engine ein »Apache« Webservice und »PHP« läuft. Anstelle von »Apache« könnte man auch einen anderen Webservice verwenden, jedoch sind andere Programme bisher nicht getestet.



EINLEITUNG

»VIVA Network-Publishing« ist eine Servertechnologie, mit deren Hilfe Anwender ohne grafische Kenntnisse komplexe Dokumente mit beliebigen Anwendungen erstellen können. Grundsätzlich kann jede Anwendung mit dem Publishing-Server zu kommunizieren, die in der Lage ist, einen HTTP-Request zu senden. Daher ist es möglich einen VIVA Network-Publishing Server über folgende Software zu steuern:

1. Eine beliebige Anwendung mit integrierter Programmiersprache
2. Einen Internet-Browser
3. Eine beliebige Programmiersprache

Zu 1: Anwendung mit integrierter Programmiersprache

Die vom Anwender eingesetzte Anwendung muss in der Lage sein, eine Textdatei mit den Daten zu erstellen und diese per HTTP-Request mit entsprechenden Parametern an einen VIVA Network-Publishing Server zu senden. Mögliche Anwendungen sind beispielsweise Microsoft Office-Programme mit der integrierten Programmiersprache Visual Basic oder Anwendungen mit einer umfangreichen Skriptsprache. Das Ergebnis dieser Anfrage (Zum Beispiel die Darstellung der erstellten Dokumentseite(n) im PDF- oder JPG-Format), kann von der Anwendung sofort weiter verarbeitet werden.

Zu 2: Internet-Browser

Innerhalb des Internet-Browser kann ein Anwender eine URL definieren, die aus Daten und Parametern besteht. Diese URL wird per HTTP-Request an einen VIVA Network-Publishing Server gesendet. Das Ergebnis dieser Anfrage (Zum Beispiel die Darstellung der erstellten Dokumentseite(n) im PDF- oder JPG-Format), wird von dem Internet-Browser sofort angezeigt.

Zu 3: Programmiersprache

Mittels einer Programmiersprache werden Daten aus einer beliebigen Datenquelle (z. B. Internetportal mit MySQL-Datenbank) exportiert und per HTTP-Request mit entsprechenden Parametern an einen VIVA Network-Publishing Server zu senden. Mögliche Programmiersprachen sind beispielsweise Visual Basic, C/C++ oder PHP in Verbindung mit einer Datenbank (z.B. MySQL). Das Ergebnis dieser Anfrage (Zum Beispiel Darstellung der Seite(n) als PDF- oder JPG-Datei), kann von der Anwendung sofort weiter verarbeitet werden.

Für alle die meisten Anwendungen finden Sie in der Dokumentation entsprechende Beispieldaten.

PARAMETERANWENDUNG

Einige beschriebene Parameter werden bereits in dem verwendeten Template eingestellt sein. Alle hier definierten Parameter überschreiben automatisch die jeweilige Einstellung im Template. Wird der jeweilige Parameter nicht definiert, wird automatisch die Defaulteinstellung, bzw. die Einstellung im Template verwendet. Die Parameterübergabe erfolgt gemäß der folgenden Syntax:

Syntax:

"&{NAME1}={WERT1}&{NAME2}={WERT2}&{NAME3}=..."

Beispiel:

&sender=application&sourceType=path&templateName=template.nwpd&dataPath=c:/data.csv

PARAMETERBESCHREIBUNG

ALLGEMEINE PARAMETER

sender

Der Parameter **sender** definiert, mit welchem Anwendungstyp die Anfrage erfolgt. Erlaubte Werte sind: **application**, **console** und **access**.

application

Bei diesem Wert erfolgt die Anfrage zur Erstellung einer Ergebnisdatei über eine beliebige Anwendung.

console

Bei diesem Wert erfolgt die Anfrage zur Erstellung einer Ergebnisdatei über die DOS-Kommandozeile.
Beispiel: `c:\php.exe c:/{VIVA-NWP-Ordner}/vivanwp.php "&name1=wert1&name2=wert2..."`

access

Bei diesem Wert erfolgt die Anfrage zur Erstellung einer Ergebnisdatei über eine Access-Datenbank. Das Verhalten ist genau wie bei **sender = application** mit der Ausnahme, dass die Datenbank zwei unterschiedliche Anfragen an den VIVA Network Publishing Server senden muss.

sourceType

Der Parameter **sourceType** definiert, wie die Daten an den VIVA Network Publishing Server übergeben werden. Erlaubte Werte sind: **fix**, **upload**, **path** und **field**. Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

fix

Bei diesem Wert müssen die Daten in der Datei "data.csv" im Verzeichnis {VIVA-NWP-Ordner}/local/data liegen.

upload

Bei diesem Wert werden die Daten mit Hilfe des Parameters **file** in einer Datei mit variablen Dateinamen übergeben. Referenz: Parameter **file**.

path

Bei diesem Wert muss mit Hilfe des Parameters **dataPath** ein absoluter Pfad auf die Datendatei übergeben werden.

field

Bei diesem Wert werden die Daten, die normalerweise in einer Datendatei (data.csv) stehen, über den Parameter **fieldData** übergeben.

Der Parameter ist auf verschiedene Weise nutzbar. Wird dort ein einzelner String übergeben, muss dieser dem Inhalt einer CSV-Datei entsprechen. Es ist jedoch ebenfalls möglich mehrere Parameter in Array-Schreibweise zu übergeben, die vom VIVA Network Publishing Server zu einem assoziativen Array zusammengefasst werden, aus dem dann die CSV-Datei generiert wird. Beispiel: `...&fieldData[Vorname]=Hans&fieldData[Nachname]=Mustermann`

Ersatzweise kann eine Schreibweise gewählt werden, die den Feldnamen ein `fieldData_` voranstellt: `...&fieldData_Vorname=Hans&fieldData_Nachname=Mustermann`

Die Parameterübergabe als Array macht es zum Beispiel möglich, direkt aus einem HTML-Formular ein Dokument zu rechnen.



dataFile

Der Parameter **dataFile** definiert die Datei, in der typischer Weise die CSV-Daten über HTTP-POST gesendet werden. Der Parameter **dataFile** wird dann benutzt, wenn beim Parameter **sourceType** der Wert **upload** verwendet wurde. Die Datendatei kann bei großen Datenmengen auch komprimiert (.gz) übertragen werden. Das Entpacken der Datendatei erfolgt automatisch durch den VIVA Network Publishing Server. Referenz: Datei »sendfile.php« im Beispielordner. Der Parameter ist für **sender** mit dem Wert **application** verfügbar.

dataPath

Der Parameter **dataPath** beschreibt den Pfad in dem sich die Daten befinden. Der Pfad muss für den VIVA Network Publishing Server erreichbar sein. Der Parameter **dataPath** wird nur dann benutzt, wenn beim Parameter **sourceType** der Wert **path** verwendet wurde. Achtung: Es dürfen nur Querstriche (/) und KEINE Backslash (\) verwendet werden.

Beispiel: c:/meinedaten/meinedatendatei.csv

Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

templateName

Name der Template-Datei, die die Formen beinhaltet. Die Template-Datei muss im Ordner: {VIVA-NWP-Ordner}/local/templates liegen. Das Dateiformat ist .vip (für alle VivaIP-Designer Versionen bis 2.3 und alle VivaDesigner Versionen bis 5.0) oder .nwpd (für alle VivaNWP-Designer ab Version 5.1)

Beispiel: template.vip (bis Version 5.x (bis März 2006)) oder template.nwpd (ab Version 5.1 (ab März 2006)). Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

templateFile

Anstatt eine hinterlegte Templatedatei zu verwenden, kann man eine Templatedatei mit dem Auftrag versenden. Der Parameter **templateFile** definiert die Templatedatei, die über HTTP-POST gesendet wird. Die Templatedatei kann bei großen Datenmengen auch komprimiert (.gz) übertragen werden. Das Entpacken der Templatedatei erfolgt automatisch durch den VIVA Network Publishing Server. Der Parameter ist für **sender** mit dem Wert **application** verfügbar.

exportName

Der optionale Parameter **exportName** definiert den Namen der Ergebnisdatei. Standardmäßig wird der Name der Datendatei (.csv-Datei) ausgegeben.

Beispiel: Wenn der Parameter **exportName** gleich Test und das Format der Ergebnisdatei ein PDF (siehe Parameter **outType**) sein soll, dann heißt die Ergebnisdatei Test.pdf.

Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

outType

Der optionale Parameter **outType** definiert das Format der Ergebnisdatei. Als Ergebnisformat akzeptiert der Server die Formate .pdf, .jpg, .png, .ps, .xml, .idml oder .nwpd. Erfolgt keine Angabe, wird standardmäßig das PDF-Format gesetzt. Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

pdf (default)

Die Ergebnisdatei wird im nativen PDF-Format (.pdf) zurückgeliefert. Das PDF-Dokument kann beliebig viele Seiten besitzen. Die Qualität des PDFs wird durch eine Reihe zusätzlicher Parameter (z. B. Parameter **quality**) definiert. Der Wert **pdf** ist auch dann zu verwenden, wenn der Acrobat Distiller angesprochen werden soll. Beachten Sie hierzu bitte den Parameter **outDevice** mit dem Wert **distiller**.



nwpd

Die Ergebnisdatei wird im nativen VivaNWP-Designer-Format (.nwpd) zurückgeliefert. Das .nwpd-Dokument kann beliebig viele Seiten besitzen und mit Hilfe des VivaDesigners oder des VivaNWP-Designers auf allen Plattformen nachbearbeitet werden. Die Ausgabe kann auch mit dem ZIP-Parameter kombiniert werden, sodass beispielsweise alle Bilder und Schriften mit dem Dokument als ZIP-Datei ausgeliefert werden.

xml

Die Ergebnisdatei wird im Viva XML-Format (.xml) zurückgeliefert. Das .xml-Dokument kann mithilfe des VivaDesigners oder des VivaNWP-Designers auf allen Plattformen nachbearbeitet werden. Im Unterschied zum nativen Dokumentformat .nwpd, können im XML keine Schriften eingebunden werden.

jpg , png, svg

Die Ergebnisdatei wird im JPEG-Format (.jpg) oder PNG-Format (.png) zurückgeliefert. Die Qualität und Größe der Ergebnisdatei wird durch eine Reihe optionaler Parameter für die Bildausgabe definiert.

ps

Die Ergebnisdatei wird im PostScript-Format zurückgeliefert (.ps). Das PostScript-Dokument kann beliebig viele Seiten besitzen. Die Qualität des PostScript-Dokuments wird durch eine Reihe zusätzlicher Parameter (z. B. Parameter **quality**) definiert.

idml

Die Ergebnisdatei wird im Adobe IDML-Format zurückgeliefert (.idml). Die Qualität des Dokuments hängt von der Exportqualität der jeweiligen Programmversion zusammen und sollte vorher getestet werden. Die Ausgabe kann auch mit dem ZIP-Parameter kombiniert werden, sodass beispielsweise alle Bilder und Schriften mit dem Dokument als ZIP-Datei ausgeliefert werden.

outHow

Der optionale Parameter **outHow** definiert, auf welche Weise die Ergebnisdatei geliefert werden soll. Zur Auswahl stehen die Werte: **relocation-javascript**, **relocation-header** und **direct**. Erfolgt keine Angabe, wird standardmäßig der Wert **relocation-javascript** gesetzt. Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

relocation-javascript (default)

Als Ergebnis wird eine HTML-Seite geliefert, die ein JavaScript beinhaltet, das auf die Ergebnisdatei verweist.

relocation-header

Auf die Ergebnisdatei wird mit dem Meta-Tag »Relocation« innerhalb des HTTP-Headers verwiesen. Verwenden Sie bei älteren Browsern, die mit dem direkten Relocation Probleme haben, den Wert **relocation-javascript**.

direct

Die Ergebnisdatei wird direkt zurückgeliefert. Bei Browseranwendungen macht daher meistens nur das Format PDF oder JPG Sinn.

outRedirectionUrl (optional)

Der optionale Parameter **outRedirectionUrl** erlaubt die Definition einer beliebigen Variable, welche innerhalb des Return den Verweis auf die Ergebnisdatei beinhaltet.

Der Parameter ist für **sender** mit dem Wert **application** verfügbar.

Beispiel: outRedirectionUrl = retURL



waitType (optional)

Der optionale Parameter **waitType** definiert das Verhalten, während die Anwendung auf die Erstellung der Ergebnisdatei wartet. Zur Auswahl stehen die Werte: **html** und **silent**. Erfolgt keine Angabe, wird standardmäßig der Wert **html** gesetzt.

html

Bei diesem Wert wird eine HTML-Seite angezeigt, die den Status des Auftrages in Form einer Fortschrittsanzeige darstellt. Die HTML-Seite wird auch dazu verwendet, mögliche Fehlermeldungen auszugeben. Daher wird die Verwendung dieses Wertes stets empfohlen. Referenz: **autoAbort**.

Der Wert ist NUR für **sender** mit dem Wert **application** verfügbar.

silent

Bei diesem Wert passiert nichts. Es wird keine Warteseite dargestellt, die den Status des Auftrages darstellt.

Der Wert ist für **sender** mit dem Wert **application** und **console** verfügbar.

waitTimeout (optional)

Der optionale Parameter **waitTimeout** definiert, wie lange die Anwendung auf die Erstellung der Ergebnisdatei warten soll, die durch den Acrobat Distiller erstellt wird. Der Parameter wirkt sich nur dann aus, wenn beim Parameter **waitType** der Standardwert **html** verwendet wird.

Beispiel: `waitTimeout = 900` (Bei diesem Beispiel wird die Wartezeit auf 900 Sekunden gesetzt)

Der Wert ist für **sender** mit dem Wert **application** und **console** verfügbar.

dstPath (optional)

Der optionale Parameter **dstPath** definiert den Ordner, in dem der VIVA Network Publishing Server die Ergebnisdatei speichern soll. Der Pfad muss lokal von dem VIVA Network Publishing Server über das Dateisystem erreichbar sein. Erfolgt keine Angabe, wird die Ergebnisdatei standardmäßig im Jobordner gesichert. Achtung: Es dürfen nur Querstriche (/) und KEINE Backslash (\) verwendet werden. Sofern Sie zur Ansteuerung des Acrobat Distiller mit Hilfe des Parameters `outDevice` die VIVA-PHP-Suite verwenden (z.B. aus VivaGate) vornehmen, hat der Parameter `dataPath` keine Auswirkungen!

Beispiel: `c:/meinkatalog/zielverzeichnis/`

Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

outDevice (optional)

Der optionale Parameter **outDevice** definiert auf welche Weise die Ergebnisdatei vom Format PDF erstellt werden soll. Die Verwendung der Variable ist nur dann sinnvoll, wenn **outType = pdf** und **quality = highres** ist. Der VIVA Network Publishing Server stellt hier verschiedene Optionen zur Verfügung. Zur Auswahl stehen die Werte: **application**, **distiller** und **printerdriver**. Erfolgt keine Angabe, wird standardmäßig der Wert **application** gesetzt.

application (default)

Bei diesem Wert wird das PDF direkt vom VIVA Network Publishing Server erzeugt. Eine solche Vorgehensweise wird immer dann empfohlen, wenn es sich um einen PDF zu Korrekturzwecken handelt. Referenz: Parameter **quality**

distiller

Bei diesem Wert wird das PDF vom „Adobe Acrobat Distiller“ erzeugt. In diesem Fall MUSS der Wert des Parameters **outType = pdf** sein. Dabei wird abweichend von den eingestellten Parametern für die Ergebnisdatei (**outType = pdf**) zunächst eine PostScript-Datei erzeugt und im Ordner `{Web-VivaIP Ordner}/HighRes/in` gesichert. Dieser Ordner muss vom »Adobe Acrobat Distiller« überwacht werden. Das



vom „Adobe Acrobat Distiller« im Out-Ordner erstellte PDF wird dann an den Absender in der gewünschten Form (siehe Parameter **outHow**) zurückgeliefert. Die Rücklieferung erfolgt nur dann erfolgreich, wenn der standardmäßige Zeitrahmen von 300 Sekunden nicht überschritten wird. Diesen Zeitfaktor kann entsprechend bestimmt werden (siehe Parameter **waitTimeout**).

printerdriver

Bei diesem Wert wird das PDF vom Druckertreiber des Betriebssystems erzeugt. Dabei wird abweichend von den eingestellten Parametern für die Ergebnisdatei (**outType** = **pdf**) zunächst eine PostScript-Datei erzeugt, die dann vom Druckertreiber in eine PDF-Datei umgewandelt wird. Das vom Druckertreiber erstellte PDF wird dann an den Absender in der gewünschten Form (siehe Parameter **outHow**) zurückgeliefert.

pagesSource (optional)

Der optionale Parameter **pagesSource** definiert, welche Dokumentseiten in der Ergebnisdatei erscheinen. Zur Auswahl stehen die Werte **job** und **document**. Erfolgt keine Angabe, wird standardmäßig der Wert **job** gesetzt. Innerhalb dieses Bereiches gelten die Parameter **firstPage** und **lastPage**. Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

job (default)

Bei diesem Wert werden sämtliche Seiten des aktuellen Produktionsauftrages als Ergebnisdatei zurückgeliefert, sofern diese nicht mit Hilfe der Parameter **firstPage** und **lastPage** eingeschränkt werden.

document

Bei diesem Wert werden sämtliche Seiten des Dokuments als Ergebnisdatei zurückgeliefert, sofern diese nicht mit Hilfe der Parameter **firstPage** und **lastPage** eingeschränkt werden.

Beispiel: Wenn das Template/Dokument vor der Anfrage 5 Dokumentseiten enthielt und während der Ausführung des Anfrags 10 neue Dokumentseiten generiert werden, dann bekommt man durch das Setzen des Parameters **pagesSource** auf den Wert **document** und des Parameters **firstPage** auf den Wert **1** und des Parameters **lastPage** auf den Wert **10** die Dokumentseiten 1 bis 10 zurück. Hätte der Parameter **pagesSource** den Wert **job**, würde man als Ergebnis die Dokumentseiten 6 bis 15 erhalten.

firstPage (optional)

Der optionale Parameter **firstPage** definiert die erste auszugebende Seite der Ergebnisdatei. Die Einstellung bezieht sich nur auf Ergebnisdateien vom Typ JPG, PS oder PDF. Erfolgt keine Angabe, wird in Abhängigkeit des Parameters **pagesSource** die erste Seite des Dokuments oder des Jobs verwendet. Mögliche Eingaben sind Ganzzahlen von 1 bis 9999. Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

lastPage (optional)

Der optionale Parameter **lastPage** definiert die letzte auszugebende Seite der Ergebnisdatei. Die Einstellung bezieht sich nur auf Ergebnisdateien vom Typ JPG, PS oder PDF. Erfolgt keine Angabe, wird in Abhängigkeit des Parameters **pagesSource** die letzte Seite des Dokuments oder des Jobs verwendet. Mögliche Eingaben sind Ganzzahlen von 1 bis 9999. Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

Beispiel: Wenn man aus dem Ergebnis nur die erste Seite erhalten möchte, setzt man den Parameter **firstPage** und **lastPage** jeweils auf **1**.

language (optional)

Der optionale Parameter **lang** definiert, in welcher Sprache Fehlermeldungen und Hinweise ausgegeben werden sollen. Zur Auswahl stehen die Werte: **de** und **en**. Erfolgt keine Angabe, wird standardmäßig der Wert **en** gesetzt.

Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

label (optional)

Der optionale Parameter **label** definiert den Präfix für den Jobticket-Ordner. Alle Jobs werden in jeweils eigenen Ordnern abgelegt. Die Ordnernamen werden automatisch generiert. Wird der Parameter **label** verwendet, wird der definierte Wert vor den automatisch generierten Ordnernamen gesetzt. Als Wert ist eine beliebige Zeichenfolge erlaubt. Die Zeichenlänge darf einschließlich des Pfadnamens 256 Zeichen nicht überschreiten.

Beispiel: Der Parameter **label** enthält den Wert **eb-** und der generierte Jobticket-Ordnername ist f9941a51eeac63dc. Der Ordner heißt anschließend eb- f9941a51eeac63dc.

Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

ticketID (optional)

Der optionale Parameter **ticketID** definiert den Namen des Jobticket-Ordners. Als Wert ist eine beliebige Zeichenfolge erlaubt. Die Zeichenlänge darf einschließlich des Pfadnamens 256 Zeichen nicht überschreiten. Erfolgt keine Angabe, erstellt die Anwendung automatisch einen Ordnernamen.

Der Parameter ist für **sender** mit dem Wert **application** und **console** verfügbar.

PARAMETER FÜR DIE AUSGABE

Alle nachfolgenden Parameter für die Ausgabe sind optional und für alle Werte des Parameters **sender** verfügbar.

print_printLayer (optional ab Version 9.1)

Der optionale Parameter **print_printLayer** definiert, welche Ebenen bei der Ausgabe berücksichtigt werden sollen. Erfolgt keine Angabe, wird standardmäßig der Wert **visible_printable** gesetzt.

Dieser optionale Parameter ist hilfreich, wenn beispielsweise bei einem LowRes-PDF andere Objekte angezeigt werden sollen, als bei einer HighRes-Ausgabe. So könnte bei einem LowRes-PDF ein Wasserzeichen oder ein Logo angezeigt werden, welches bei der HighRes nicht angezeigt werden soll, da es sich um einen Eindruck auf einem Vordruck handelt.

visible_printable (Standardwert)

Bei diesem Wert werden sowohl alle sichtbaren als auch alle druckbaren Ebenen ausgegeben.

visible

Bei diesem Wert werden ausschließlich alle sichtbaren Ebenen ausgegeben.

printable

Bei diesem Wert werden ausschließlich alle druckbaren Ebenen ausgegeben.

all

Bei diesem Wert werden alle Ebenen ausgegeben, unabhängig davon, ob sie druckbar sind oder nicht.

print_imageQuality (optional)

Der optionale Parameter **print_imageQuality** definiert, ob die Ergebnisdatei vom Format PDF oder PS mit einer hohen (highRes) oder niedrigen (lowRes) Bildauflösung erstellt werden soll. Zur Auswahl stehen die Werte **highres** oder **lowres**. Erfolgt keine Angabe, wird standardmäßig der Wert **lowres** gesetzt.

highres

Bei diesem Wert wird die original Bilddatei ausgegeben. Der Wert **highres** wird in der Regel verwendet, wenn es sich bei der Ergebnisdatei um ein PDF oder ein PS-Dokument handelt, welches auf einem hochauflösenden Geräte ausgegeben werden soll.

lowres

Bei diesem Wert wird nur die Bildschirmdarstellung der Bilddatei (72 DPI-Vorschau des Bildes) ausgegeben. Der Wert **lowres** wird in der Regel verwendet, wenn es sich bei der Ergebnisdatei um ein PDF handelt, das nur zu Korrekturzwecken benötigt wird.

noimage

Bei diesem Wert werden keine Bilder ausgegeben. Der Wert **noimage** wird in der Regel verwendet, wenn es sich bei der Ergebnisdatei um ein PDF handelt, das nur zu Korrekturzwecken benötigt wird.

Marken (optional)

Der optionale Parameter **print_cropMarks** definiert, ob die Ergebnisdatei mit Beschnittmarken erstellt wird. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_cropMarkLength** definiert die Länge der Beschnittmarken. Als Wert wird ein **String** akzeptiert.(z.B. »3 mm«).

Der optionale Parameter **print_cropMarkDistance** definiert den Abstand der Beschnittmarken zur Seite. Als Wert wird ein **String** akzeptiert.(z.B. »3 mm«).

Der optionale Parameter **print_registrationMarks** definiert, ob die Ergebnisdatei mit Passkreuzen erstellt wird. Als Wert wird **true** oder **false** akzeptiert.

Anschnitt (optional)

Der optionale Parameter **print_useBleed** definiert, ob die Ergebnisdatei mit einem Anschnitt erstellt wird. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_bleed** definiert die Größe des Anschnitts. Alternativ kann man auch für jeden Seitenrand mit den Parametern **print_bleedLeft**, **print_bleedTop**, **print_bleedRight** und **print_bleedBottom** den Anschnitt definieren. Als Wert wird ein **String** akzeptiert.(z.B. »3 mm«).

PDF Optimierungen (optional ab Version 9.5)

Der optionale Parameter **print_pdfLinearize** definiert, ob die Ergebnisdatei für das Web optimiert werden soll. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_pdfCropImages** definiert, ob die Bilder in der Ergebnisdatei beschnitten werden sollen. Dadurch kann unter Umständen die Dateigröße verkleinert werden. Als Wert wird **true** oder **false** akzeptiert.



Der optionale Parameter **print_pdfIncludeICCProfiles** definiert, ob das Farbprofil in die Ergebnisdatei eingebunden werden sollen. Dadurch erhöht sich auch die Dateigröße. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_FlattenTransparency** definiert, ob Transparenzen in Bitmaps umgerechnet werden sollen. Dadurch erhöht sich unter Umständen die Dateigröße. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_pdfFlattenTransparencyResolution** definiert die Auflösung, in der die Transparenzen in Bitmaps umgerechnet werden. Als Wert werden Ganzzahlen zwischen 36 und 500 akzeptiert.

Der optionale Parameter **print_pdfRasterizeVectorPictures** definiert, ob Transparenzen von Vektorbildern in Bitmaps umgerechnet werden sollen. Dadurch erhöht sich unter Umständen die Dateigröße. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_pdfRasterizeVectorPicturesResolution** definiert die Auflösung, in der die Transparenzen von Vektorbildern in Bitmaps umgerechnet werden. Als Wert werden Ganzzahlen zwischen 36 und 500 akzeptiert.

Der optionale Parameter **print_pdfReduceImageResolution** definiert, ob die Auflösung von Bildern reduziert werden soll. Dadurch kann unter Umständen die Dateigröße verkleinert werden. Als Wert wird **none**, **preflight preferences** oder **custom** akzeptiert.

Der optionale Parameter **print_pdfMaxColorImageResolution** definiert die Auflösung von Farbbildern, wenn beim Parameter **print_pdfReduceImageResolution** der Wert **custom** verwendet wurde. Als Wert werden Ganzzahlen zwischen 36 und 2.000 akzeptiert.

Der optionale Parameter **print_pdfMaxGrayImageResolution** definiert die Auflösung von Graustufenbildern, wenn beim Parameter **print_pdfReduceImageResolution** der Wert **custom** verwendet wurde. Als Wert werden Ganzzahlen zwischen 36 und 2.000 akzeptiert.

Der optionale Parameter **print_pdfMaxMonoImageResolution** definiert die Auflösung von Schwarz-/Weißbildern, wenn beim Parameter **print_pdfReduceImageResolution** der Wert **custom** verwendet wurde. Als Wert werden Ganzzahlen zwischen 36 und 2000 akzeptiert.

PDF Optionen(optional)

Der optionale Parameter **print_pdfCompatibility** definiert das Format der PDF-Datei. Als Wert wird **PDF 1.3**, **PDF 1.4**, **PDF 1.5**, **PDF 1.6**, **PDF 1.7**, **PDF 1.7 extension level 3**, **PDF 1.7 extension level 8**, **PDF 2.0**, **PDF/X-1a:2003**, **PDF/X-3:2003**, **PDF/X-4**, **PDF/X-4p**, **PDF/X-5g**, **PDF/X-5pg** akzeptiert.

Der optionale Parameter **print_pdfOptimize** definiert, ob die PDF-Datei optimiert werden soll. Als Wert wird **true** oder **false** akzeptiert. Der Parameter darf ab Version 9.5 nicht mehr verwendet werden.

Der optionale Parameter **print_pdfCompressionMethod** definiert das Kompressionsverfahren der PDF-Datei. Als Wert wird **none**, **ZIP** oder **JPEG** akzeptiert.

Der optionale Parameter **print_pdfCompressionQuality** definiert die Qualität für das jeweilige Kompressionsverfahren. Als Wert werden Prozentwerte von **0** bis **100** akzeptiert.



Der optionale Parameter **print_pdfCompressText** definiert, ob der Text in der PDF-Datei komprimiert werden soll. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_pdfWarningScale** definiert, ob eine Warnung für die Bildskalierung in der PDF-Datei ausgegeben werden soll. Als Wert wird **true** oder **false** akzeptiert.

Die optionalen Parameter **print_pdfWarningScaleBelow** und **print_pdfWarningScaleAbove** definieren die Minimum- bzw. Maximumskalierung für Bilder in Prozent (%). Werden diese Werte unter- oder überschritten wird im PDF eine Notiz (sticky note) erzeugt, in der die Skalierungswerte stehen. Als Wert werden Prozentwerte von **0** bis **100** akzeptiert. Der Parameter wird nur ausgewertet, wenn der Parameter **print_pdfWarningScale** den Wert **true** besitzt.

Der optionale Parameter **print_pdfWarningTextOverflow** definiert, ob eine Warnung für Textüberlauf in der PDF-Datei ausgegeben werden soll. In diesem Fall wird eine Notiz (sticky note) erzeugt, die die ersten 200 Zeichen des Übersatzes enthält. Als Wert wird **true** oder **false** akzeptiert.

ICC-Profile (optional)

Version 8:

Die optionalen Parameter **print_pdfxOutputIntent**, **print_pdfxGrayProfile**, **print_pdfxRGBProfile** und **print_pdfxCMYKProfile** bestimmen die zu verwendenden ICC-Profile. Als Wert wird ein String akzeptiert, der dem Dateinamen des Profils entspricht (z.B. »CoatedFOGRA27.icc«) ohne Angabe eines Pfades. Das Farbprofil muss sich dabei im Programmordner »Resources/Color Profiles« befinden, da Profile NICHT in das native VIVA-Format eingebettet werden.

Version 9:

Der optionale Parameter **print_pdfColorConversion** bestimmt, ob die Farbkonvertierung vorgenommen werden soll. Als Wert wird **none**, **always** oder ab Version 9.5 **preserve numbers** akzeptiert.

Der optionale Parameter **print_pdfTargetProfile** bestimmt das zu verwendende ICC-Profile. Als Wert wird ein String akzeptiert, der dem Dateinamen des Profils entspricht (z.B. »CoatedFOGRA27.icc«) ohne Angabe eines Pfades. Das Farbprofil muss sich dabei im Programmordner »Resources/Color Profiles« befinden, da Profile NICHT in das native VIVA-Format eingebettet werden.

Hinweis:

- Die optionalen Parameter **print_pdfxOutputIntent** und **print_pdfxGrayProfile** bestimmen die zu verwendenden ICC-Profile bei der PDF/X-Ausgabe.
- Die Parameter **print_pdfxRGBProfile** und **print_pdfxCMYKProfile** dürfen ab Version 9.0 nicht mehr verwendet werden!
- Der Parameter **print_pdfxGrayProfile** darf ab Version 9.5 nicht mehr verwendet werden!

Parameter für JPG, PNG, Bildausgabe (optional)

Der optionale Parameter **print_outputResolution** definiert die Auflösung (Maßeinheit ppi), in der die Ergebnisdatei ausgegeben werden soll. Als Wert werden Ganzzahlen zwischen 1 und 1.000 akzeptiert.

Der optionale Parameter **print_outputWidth** definiert die Breite (Maßeinheit px), in der die Ergebnisdatei ausgegeben werden soll. Als Wert werden Ganzzahlen zwischen 1 und 100.000 akzeptiert. Der Wert »0« (Default) entspricht einer automatischen Breite auf Basis des Dokumentformates.



Der optionale Parameter **print_outputHeight** definiert die Höhe (Maßeinheit px), in der die Ergebnisdatei ausgegeben werden soll. Als Wert werden Ganzzahlen zwischen 1 und 100.000 akzeptiert. Der Wert »0« (Default) entspricht einer automatischen Höhe auf Basis des Dokumentformates.

Der optionale Parameter **print_outputQuality** definiert die Qualität (Maßeinheit %), in der die Ergebnisdatei ausgegeben werden soll. Als Wert werden Ganzzahlen zwischen 1 und 100 akzeptiert.

Diverse Parameter (optional)

Der optionale Parameter **print_pageNumbering** definiert die Seitennummerierung in der PDF-Datei. Als Wert wird **Physical**, **Logical** oder **PhysicalAndLogical** akzeptiert.

Die optionalen Parameter **print_horizontalAlignment** und **print_verticalAlignment** definieren die Ausrichtung des Dokuments. Die Verwendung dieses Parameters ist nur sinnvoll, wenn das im Dokument gespeicherte Bogen oder Rollenformat (Papierformat) größer ist als das Dokumentformat (z.B. A5-Seitenformat auf A4-Papierformat ausgeben). Als Wert werden **left**, **center** oder **right** bzw. **top**, **center** oder **bottom** akzeptiert.,

Der optionale Parameter **print_fontEmbedding** definiert, ob und wie Schriften in der PDF-Datei eingebettet werden sollten. Als Wert wird **none**, **type1 but standard 14**, **type1** oder **all** akzeptiert.

Der optionale Parameter **print_printSpreads** definiert, ob nebeneinanderliegende Seiten als Bogen ausgegeben werden sollten. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_ignoreTrapping** definiert, ob die automatische Farbüberfüllung angewendet werden soll. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_printSpellCorrection** definiert, ob die Anzeige von Rechtschreibfehlern in der Ergebnisdatei angezeigt werden soll. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **print_opi** definiert, ob Bilder auf einem OPI-Server für die Ausgabe gesucht werden sollen. Als Wert wird **none**, **TIFFOnly**, **EPSOnly**, **TIFFAndEPS** oder **auto** akzeptiert.

colorSeparation (optional)

Der optionale Parameter **colorSeparation** definiert, ob die Ergebnisdatei farbsepariert werden soll. Bei der Farbseparation wird für jede verwendete Prozess- und Schmuckfarben auf einer Dokumentseite eine eigene Farbseite erstellt. Zur Auswahl stehen die Werte **off**, **application** und **inrip**. Erfolgt keine Angabe, wird standardmäßig der Wert **off** gesetzt. Die Verwendung des Parameters ist nur dann sinnvoll, wenn der Parameter **outType** den Wert **pdf** oder **ps** besitzt. Bei einer gewünschten Farbseparation empfehlen wir dringend, die Funktion über ein RIP ausführen zu lassen, da hier neben einer professionellen Farbseparation (selbst von RGB-Bildern) auch eine hochwertige Berechnung von Überfüllungen erfolgt.

off (default)

Bei diesem Wert erfolgt keine Farbseparation. Die Ausgabe der Ergebnisdatei erfolgt im „Composite-Modus“.

application

Bei diesem Wert wird die Farbseparation direkt vom VIVA Network Publishing Server erstellt. Hierbei ist zu beachten, dass es unter Umständen zu Fehlern kommen kann, wenn als Ergebnisdatei das PDF-Format (Parameter **outType**) verwendet wird und das erzeugte Dokument Vektor-EPS-Dateien oder PDF-



Dateien besitzt. Wird als Ergebnisdatei das PS-Format (Parameter **outTyp**) verwendet, treten in diesem Bereich keine Probleme auf.

inrip

Bei diesem Wert wird die Farbseparation nicht vom VIVA Network Publishing Server, sondern von einem RIP vorgenommen. Als Ergebnisdatei liefert der Parameter unabhängig vom gewünschten Format der Ergebnisdatei (Parameter **outTyp**) eine PostScript-Datei mit entsprechenden Kommandos für die InRIP-Farbseparation. Bitte beachten Sie hierzu die Optionen des verwendeten RIPs.

ZIP Parameter (optional)

Der optionale Parameter **service_includeDocument** definiert, ob bei der Erstellung der ZIP-Datei das Dokument einbezogen werden soll. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **service_includeProtocol** definiert, ob bei der Erstellung der ZIP-Datei das Dokumentprotokoll einbezogen werden soll. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **service_includeFonts** definiert, ob bei der Erstellung der ZIP-Datei die Schriften einbezogen werden sollen. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **service_includePictures** definiert, ob bei der Erstellung der ZIP-Datei die Bilder einbezogen werden sollen. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **service_searchInSystemFontFolder** definiert, ob bei der Erstellung der ZIP-Datei Schriften im Systemordner gesucht werden sollen. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **service_searchInSpecialFontFolder** definiert, ob bei der Erstellung der ZIP-Datei Schriften in einem speziellen Ordner gesucht werden sollen. Als Wert wird **true** oder **false** akzeptiert.

Der optionale Parameter **service_specialFontFolder** definiert den Pfad zum speziellen Schriftenordner. Als Wert wird **ein String** akzeptiert.

Der optionale Parameter **service_useSubfolders** definiert, ob für Schriften und Bilder jeweils Unterordner angelegt werden sollen. Als Wert wird **true** oder **false** akzeptiert.



ANWENDUNGSBEISPIELE

VIVA hat eine Reihe von Beispielen erstellt, die aus unterschiedlichen Anwendungen eine Visitenkarte erstellen. Diese Beispiele liefert VIVA im Quellcode aus und dienen dem Entwickler als Vorlage für eigene Programme und zum besseren Verständnis der Technologie.

Ansteuerung über eine Programmiersprache

1. PHP

Sie können einen »VIVA Network-Publishing Server« sehr komfortabel über »PHP« steuern. Eine ausführliche Demoanwendung ist Bestandteil dieser Dokumentation.

2. JAVA

Sie können einen »VIVA Network-Publishing Server« sehr komfortabel über »JAVA« steuern. Eine ausführliche Demoanwendung ist Bestandteil dieser Dokumentation.

3. BORLAND DELPHI

Sie können einen »VIVA Network Publishing Server« auch über »Borland Delphi« steuern. Eine ausführliche Demoanwendung ist Bestandteil dieser Dokumentation.

Ansteuerung über eine Datenbank

1. Microsoft Access

Sie können einen »VIVA Network Publishing Server« auch über eine »Microsoft Access« Datenbank steuern. Eine ausführliche Demoanwendung ist derzeit in Arbeit und wird anschließend Bestandteil dieser Dokumentation.

Ansteuerung über einen Internet-Browser

Sie können einen »VIVA Network Publishing Server« auch direkt über ein URL steuern, die man in einem Internetbrowser eingibt. Der Syntax zur Ansteuerung eines VIVA Network Publishing Servers besteht aus drei Bereichen:

{VIVA NETWORK-PUBLISHING SERVER}/{SCRIPT}?{PARAMETER}

{VIVA NETWORK-PUBLISHING SERVER}

Die Adresse des VIVA Network-Publishing Servers

Beispiel: <http://viva-nwp.de> oder <http://196.168.0.20>

{SCRIPT}

Name des Skripts, das die Anfrage startet

Beispiel: [viva-nwp.php](#)

{PARAMETER}

Parameter, die an {VIVAIPSCRIPT} übergeben werden. Die Parameterliste beginnt mit einem Fragezeichen (?). Der Syntax besteht aus dem Parameternamen und dem entsprechenden Wert. Es können beliebig viele Parameter angegeben werden, die jeweils durch ein &-Zeichen voneinander getrennt werden.

Syntax: ?{NAME1}={WERT1}&{NAME2}={WERT2}&{NAME3}=...

Beispiel: [?sender=application&sourceType=fix&template_name=template.vip](#)

Nachfolgend ein Beispiel für eine Anwendung, die eine Anfrage an den VIVA Network Publishing Server mit der Aufgabe sendet, ein Dokument mit dem PDF-Format zu erstellen, das aus der zweiten bis fünften Dokumentseite besteht:

<http://viva-nwp-server/viva-nwp.php?sender=application&sourceType=fix&templateName=template.nwpd&outType=pdf&quality=lowres&firstPage=2&lastPage=5>

ANSTEUERUNG ÜBER DIE DOS-KOMMANDOZEILE

Der Syntax zur Ansteuerung eines VIVA Network Publishing Servers über die DOS-Kommandozeile besteht aus drei Bereichen:

Im Unterschied zum Web/Applikation muss die Ausführung auf dem Server laufen (oder woher kennt der Rechner die Adresse??

{PROGRAMM} {SKRIPT} {PARAMETER}

{PROGRAMM}

Aufruf des PHP-Programms

Beispiel c:\php4\php.exe

{SCRIPT}

Name des Scripts das die Anfrage startet, wobei hier der vollständige Pfad angegeben werden muss. Es dürfen nur Querstriche (/) und KEINE Backslash (\) verwendet werden.

Beispiel: c:/websites/kunde/vivanwp.php

{PARAMETER}

Parameter, die an {SCRIPT} übergeben werden. Die Parameterliste beginnt mit einen (hier &-Zeichen??) Fragezeichen (?). Der Syntax besteht aus dem Parameternamen und dem entsprechenden Wert. Es können beliebig viele Parameter angegeben werden, die jeweils durch ein »&«-Zeichen voneinander getrennt werden.

Syntax: ?{NAME1}={WERT1}&{NAME2}={WERT2}&{NAME3}=...

Beispiel: ?sender=application&sourceType=fix&template_name=template.vip

{PARAMETER} Parameter, die an "vivaip.php" Skript (immer vivanwp.php?) übergeben werden.

Syntax: &{NAME1}={WERT1}&{NAME2}={WERT2}&{NAME3}=...

Beispiel: &sender=console&sourceType=path&template_name=template.vip

&scType=online&dataPath=c:/daten.csv&dst_path=c:/destination/"

Beispiel: Erzeugung von einem Gesamt-PDF:

c:\php4\php.exe c:/websites/softcarrier/vivaip.php "&sender=console

&sourceType=path

&templateName=Template_Neutral.vip

&scType=pdf

&dataPath=c:/daten.csv

&dstPath=c:/destination/

&highRes=0

&cropmarks=0"



GEÄNDERTE VARIABLENNAMEN

template_name - siehe templateName.
dst_name - siehe dstPath.
cropmarks - siehe cropMarks.

GEÄNDERTE VARIABLENANWENDUNG

distiller - benutze outType = distiller.

OBSOLETE FUNKTIONEN

outRedirectionUrl (optional)

Der optionale Parameter **outRedirectionUrl** erlaubt die Definition einer beliebigen Variable, welche innerhalb des Return den Verweis auf die Ergebnisdatei beinhaltet.

Der Parameter ist für **sender** mit dem Wert **application** verfügbar.

Beispiel: outRedirectionUrl = retURL