

Technical Documentation for addressing a VIVA Network Publishing Server for Windows, Mac OS and Linux Version 3.1 - Status: Monday, August 15, 2016

DEFINITION	1	firstPage (optional).....	7
INTRODUCTION	1	lastPage (optional).....	7
USING PARAMETERS	2	language (optional).....	7
PARAMETER DESCRIPTION	3	label (optional).....	7
GENERAL PARAMETERS	3	ticketID (optional).....	7
sender.....	3	PRINTING PARAMETERS	8
sourceType.....	3	print_printLayer (optional).....	8
dataFile.....	3	print_imageQuality (optional).....	8
dataPath.....	3	Marks (optional).....	8
templateName.....	4	Bleed (optional).....	9
templateFile.....	4	PDF Options(optional).....	9
exportName.....	4	ICC Profiles (optional).....	9
outType.....	4	Various Parameters (optional).....	10
outHow.....	5	colorSeparation (optional).....	10
outRedirectionUrl (optional).....	5	ZIP Parameter (optional).....	10
waitType (optional).....	5	APPLICATION EXAMPLES	11
waitTimeout (optional).....	6	Addressing via a programming language.....	12
dstPath (optional).....	6	Addressing via a database.....	12
outDevice (optional).....	6	Addressing via an Internet Browser.....	12
pagesSource (optional).....	6	ADDRESSING VIA DOS COMMANDS	12

Copyright

© 1994-2016 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

This documentation describes the communication between an application and the »VIVA Network Publishing Server«. It does not describe the basic functionality and way of working of the »VIVA Network Publishing Server«, which will be explained in another document.

In this documentation the following terms will be used:

1. Request
By Request we refer to the transfer of the Data and Parameters to the »VIVA Network Publishing Server«, with the command to create the appropriate documents.
2. Data
By Data we mean the information which is to be processed by the »VIVA Network Publishing Server« (Data File, Parameters, etc.).
3. Parameters
By parameters we mean the information that tells the »VIVA Network Publishing Server« in which format and with which settings (value) the resulting file should be created.
4. Data File
By Data File we mean the text file (.csv format) which contains the information to be displayed. Images are represented in the form of names and possibly folders. Large Data Files can also be transferred compressed in the GZIP format (.gz).
5. Result file
By Result File we mean the document that the »VIVA Network Publishing Server« creates, based on the Data File and Parameters.
6. Template
A model that contains all the information and rules that define how the Data should be processed and set out graphically. A template is a native »VivaNWP Designer« file.
7. VivaNWP-Server (VIVA Network Publishing Server)
A »VivaNWP-Server« is a Publishing Server on which an »Apache« Webservice and »PHP« run in addition to the VivaNWP Publishing Engine. Instead of »Apache« another Webservice could be applied, but to date no other programs have been tested.



INTRODUCTION

»VIVA Network Publishing« is a Server technology with which users without graphic knowledge can create complex documents with any application. Every application can in principle communicate with the Publishing Server, providing it is able to send an HTTP request. Therefore it is possible to run a VIVA Network Publishing Server through the following software:

1. Any application with an integrated programming language
2. An Internet Browser
3. Any programming language

To 1: Application with an integrated programming language

The application applied by the user must be able to create a text file with the Data and send this via HTTP request with the corresponding parameters to a »VIVA Network Publishing Server«. Possible applications are for example Microsoft Office programs with the integrated programming language Visual Basic or applications with an extensive script language. The result of this request (for example display of the document page(s) created in PDF or JPG format) can with immediate effect be further edited by the application.

To 2: Internet Browser

In the Internet Browser a user can define a URL which consists of Data and Parameters. This URL is sent by HTTP request to a »VIVA Network Publishing Server«. The result of this request (for example display of the document page(s) created in PDF or JPG format) is displayed immediately by the Internet Browser.

To 3: Programming language

With the help of a programming language, Data from any source (e. g. Internet portal with MySQL Database) are exported and sent by HTTP request with corresponding parameters to a »VIVA Network Publishing Server«. Possible programming languages are for example Visual Basic, C/C++ or PHP in connection with a database (e.g. MySQL). The result of this request (for example display of the document page(s) created in PDF or JPG format) can with immediate effect be further edited by the application.

For most of the applications you will find corresponding example files in the documentation.

USING PARAMETERS

Some of the parameters described will already be set up in the template. All the parameters defined here automatically overwrite the appropriate setting in the template. If the relevant parameter is not defined, the default setting or the default setting in the template will be used. The transfer of parameters is achieved using the following syntax:

Syntax:

"&{NAME1}={VALUE1}&{NAME2}={VALUE2}&{NAME3}=..."

Example:

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

PARAMETER DESCRIPTION

GENERAL PARAMETERS

sender

The parameter **sender** defines with which application type the request is made. The terms permitted are: **application**, **console** and **access**.

application

With this value the request is made to create a Result File via a selected application.

console

With this value the request is made to create a Result File via a DOS comand.

Example: c:\php.exe c:/\{VIVA-NWP-Folder}/vivanwp.php "&name1=value1&name2=value2..."

access

With this value the request is made to create a Result File via an »Access database«. The process is exactly the same as with **sender** = **application** with the exception that the database must send two different requests to the VIVA Network Publishing Server.

sourceType

The parameter **sourceType** defines how the Data should be transferred to the VIVA Network Publishing Server. The terms permitted are: **fix**, **upload**, **path** und **field**. The parameter is available for **sender** with the values **application** and **console**.

fix

With this value, the Data in the file "data.csv" must lie in the folder {VIVA-NWP-Folder}/local/data.

upload

With this value the Data are transferred using the parameters **file** to a file with variable file names. Reference: parameter **file**.

path

With this value, an absolute path must be transferred to the Data File using the parameter **dataPath**.

field

With this value the Data that are normally found in a Data File (data.csv) are transferred using the parameter **fieldData**.

dataFile

The parameter **dataFile** defines the file, typically CSV files, that are sent via HTTP-POST. The parameter **dataFile** only comes into use when in the parameter **sourceType** the value **upload** was used. Large data Files can also be transferred as compressed files (.gz). The VIVA Network Publishing Server will unpack the Data File automatically. Reference: File »sendfile.php« in the Example folders. The parameter is available for **sender** with the value **application**.

dataPath

The parameter **dataPath** describes the path in which the Data are to be found. The path must be accessible for the VIVA Network Publishing Server. The parameter **dataPath** only comes into use when in the parameter **sourceType** the value **path** was used. Attention: Only Slashes (/) and NO Backslash (\) may be used.



Example: c:/mydata/mydatafile.csv

The parameter is available for **sender** with the values **application** and **console**.

templateName

Name of the template file containing the forms. The template file must lie in the folder: {VIVA-NWP-Folder}/local/templates. The file format is .vip (for all VivaIP Designer versions to 2.3 and all VivaDesigner versionen to 5.0) or .nwpd (for all VivaNWP Designer from version 5.1)

Example: template.vip (to version 5.x (to March 2006)) or template.nwpd (from version 5.1 (from March 2006)). The parameter is available for **sender** with the values **application** and **console**.

templateFile

Instead of using a template file from the server, it is possible to send a template file with the order. The parameter **templateFile** defines the file that is sent via HTTP-POST. Large data Files can also be transferred as compressed files (.gz). The VIVA Network Publishing Server will unpack the template file automatically. The parameter is available for **sender** with the value **application**.

exportName

The optional parameter **exportName** defines the name of the Result File. By default the name of the Data File (.csv-Datei) will be applied.

Example: If the parameter **exportName** is Test and the format of the Result File is to be a PDF (see parameter **outType**), then the Result File will be Test.pdf.

The parameter is available for **sender** with the values **application** and **console**.

outType

The optional parameter **outType** defines the format of the Result File. As result format the Server accepts the formats .pdf, .jpg, .png, .ps, .xml, .idml or .nwpd. If no entry is made, the PDF-Format will be set as default. The parameter is available for **sender** with the values **application** and **console**.

pdf (default)

The Result File will be delivered in native PDF format (.pdf). The PDF document can have an unlimited number of pages. The quality of the PDF is defined through a series of additional parameters (e. g. parameter **quality**). The value **pdf** must even be used when the Acrobat Distiller is addressed. Please refer to the parameter **outDevice** with the value **distiller**.

nwpd

The Result File will be delivered in native VivaNWP Designer format (.nwpd). The .nwpd document can have an unlimited number of pages and may be edited further using VivaDesigner or VivaNWP Designer on all platforms. The output can also be combined with ZIP parameters, so that for example all images and fonts are delivered with the document as a ZIP file.

xml

The Result File will be delivered in VivaXML format (.xml). The .xml document may be edited further using VivaDesigner or VivaNWP Designer on all platforms. By contrast to the native format .nwpd, no fonts can be embedded in the XML.

jpg or png

The Result File will be delivered in JPEG format (.jpg) or PNG format (.png).

ps

The Result File will be delivered in PostScript format (.ps). The PostScript document can have an unlimited number of pages. The quality of the PostScript document is defined through a series of additional parameters (e. g. parameter **quality**).

idml

The Result File will be delivered in Adobe IDML-Format (.idml). The document quality depends on the export quality of the appropriate program version and should be tested beforehand. The output can also be combined with the ZIP parameters, so that for example all images and fonts are delivered with the document as a ZIP file.

outHow

The optional parameter **outHow** defines in which way the Result File should be delivered. There is a choice of the following values: **relocation-javascript**, **relocation-header** and **direct**. If no entry is made, the value **relocation-javascript** will be applied as default. The parameter is available for **sender** with the values **application** and **console**.

relocation-javascript (default)

As a result an HTML page will be delivered containing a JavaScript that leads to the Result File.

relocation-header

The Meta-Tag »Relocation« inside the HTTP header leads to the Result File. For older Browsers, that have problems with direct Relocation, use the value **relocation-javascript**.

direct

The Result File will be delivered directly. With Browser applications generally only the formats PDF or JPG make any sense.

outRedirectionUrl (optional)

The optional parameter **outRedirectionUrl** enables the definition of any variable that contains the reference to the Result file within the Return.

the parameter is available for **sender** with the value **application**.

Example: outRedirectionUrl = retURL

waitType (optional)

The optional parameter **waitType** defines the behavior while the application is waiting for the creation of the Result File. There is a choice of values: **html** and **silent**. If no entry is made, the value **html** will be applied as default.

html (default)

With this value an HTML page will be displayed showing the status of the order in the form of a progress indicator. The HTML page is also used to issue possible error warnings. Therefore it is always recommended to use this value. Reference: **autoAbort**

The value is ONLY available for **sender** with the value **application**.

silent

With this value nothing happens. No interim page will be displayed to show the status of the order.

The value is available for **sender** with the values **application** and **console**.



waitTimeout (optional)

The optional parameter **waitTimeout** defines how long the application should wait for the creation of the Result File that is created by Acrobat Distiller. The parameter only has an effect if in the parameter **waitType** the default value **html** is set.

Example: waitTimeout = 900 (In this example the waiting time is set to 900 seconds)

The value is available for **sender** with the value **application**.

dstPath (optional)

The optional parameter **dstPath** defines the folder in which the »VIVA Network Publishing Server« should save the Result File. The path must be accessible by the »VIVA Network Publishing Server« via the file system. If no entry is made, the Result File will be saved in the job folder. Attention: Only Slashes (/) and NO Backslash (\) may be used. If you address the Acrobat Distiller using the parameter outDevice with the VIVA PHP suite (e.g. from VivaGate), the parameter **dataPath** will have no effect!

Example: c:/mycatalog/destinationfolder/

The parameter is available for **sender** with the values **application** and **console**.

outDevice (optional)

The optional parameter **outDevice** in which way the Result File in PDF format should be created. The use of this variable only makes sense when **outType** = **pdf** and **quality** = **highres**. The »VIVA Network Publishing Server« offers differing options here. There is a choice between the values: **application**, **distiller** and **printerdriver**. If no entry is made, the value **application** will be applied as default. The parameter is available for **sender** with the values **application** and **console**.

application (default)

With this value, the PDF will be created directly by the VIVA Network Publishing Server. Such a procedure is always recommended when PDFs for proofing are created. Reference: parameter **quality**

distiller

With this value, the PDF will be created by "Adobe Acrobat Distiller". In this case, the value of the parameter MUST be **outType** = **pdf**. Here, departing from the parameters set for the resulting file (**outType** = **pdf**) a PostScript file will first be created and saved in the folder {Web-VivaIP Ordner}/HighRes/in. This folder must be watched over by "Adobe Acrobat Distiller". The PDF created by "Adobe Acrobat Distiller" in the Out folder will then be delivered back to the requestor in the appropriate form (see parameter **outHow**). The Return is only successful if the default time frame of 300 seconds is not exceeded. This time factor can be defined accordingly (see Parameter **waitTimeout**).

printerdriver

With this value, the PDF will be created by the printer driver of the operating system. Here, departing from the parameters set for the resulting file (**outType** = **pdf**) a PostScript file will first be created, which will then be converted to a PDF file by the printer driver. The PDF created by the printer driver will then be delivered back to the requestor in the appropriate form (see parameter **outHow**).

pagesSource (optional)

The optional parameter **pagesSource** defines which document pages should appear in the Result File. There is a choice between the values **job** and **document**. If no entry is made, the value **job** will be applied as default. Within this sector are the parameters **firstPage** und **lastPage**. The parameter is available for **sender** with the values **application** and **console**.

job (default)

With this value, all the pages of the current production order will be delivered as the Result File, insofar



as these have not been limited using the parameters **firstPage** and **lastPage**.

document

With this value, all the pages of the document will be delivered as the Result File, insofar as these have not been limited using the parameters **firstPage** and **lastPage**.

Example: If the template/document contained 5 pages before the request and in complying with the request 10 new document pages are generated, then by setting the parameter **pagesSource** to the value **document**, the parameter **firstPage** to the value **1** and the parameter **lastPage** to the value **10**, the document pages 1 to 10 will be returned. If the parameter **pagesSource** would be set to the value **job**, then one would receive as a result the document pages 6 to 15.

firstPage (optional)

The optional parameter **firstPage** defines the first page of the Result File to be output. The setting relates only to Result Files of the types JPG, PS or PDF. If no entry is made, the first page of the document or the job will be applied according to the parameter **pagesSource**. Possible entries are whole numbers from 1 to 9999.

The parameter is available for **sender** with the values **application** and **console**.

lastPage (optional)

The optional parameter **lastPage** defines the last page of the Result File to be output. The setting relates only to Result Files of the types JPG, PS or PDF. If no entry is made, the last page of the document or the job will be applied according to the parameter **pagesSource**. Possible entries are whole numbers from 1 to 9999.

The parameter is available for **sender** with the values **application** and **console**.

Example: If you only want to receive the first page of the result, you set both parameters **firstPage** and **lastPage** to **1**.

language (optional)

The optional parameter **lang** defines in which language error warnings and information should be output. There is a choice between the values: **de** and **en**. If no entry is made, the value **en** will be applied as default.

The parameter is available for **sender** with the values **application** and **console**.

label (optional)

The optional parameter **label** defines the prefix for the Job Ticket folder. All jobs are placed in their own folders. The folder names are generated automatically. If the parameter **label** is applied, the value defined will be placed before the automatically generated folder names. Any series of characters is allowed as a value. The length including the path name may not exceed 256 characters.

Example: The parameter **label** contains the value **eb-** and the generated Job Ticket folder name is f9941a51eeac63dc. The folder is then called eb- f9941a51eeac63dc.

The parameter is available for **sender** with the values **application** and **console**.

ticketID (optional)

The optional parameter **ticketID** defines the name of the Job Ticket folder. Any series of characters is allowed as a value. The length including the path name may not exceed 256 characters. If no entry is made, the application will create a folder name automatically.



The parameter is available for **sender** with the values **application** and **console**.

PRINTING PARAMETERS

All the following printing parameters are optional and available for all values of the parameter **sender**.

print_printLayer (optional from Version 9.1)

The optional parameter **print_printLayer** defines which layers will be printed/exported. If no entry is made, the default value **visible_printable** will be set.

This optional parameter is helpful if for example other objects should be shown in a LowRes PDF than in a High-Res output. Thus in a Low-Res PDF a watermark or a logo could be displayed that should not be output in High-Res, as it is only there as an impression of a pre-print (e.g. letterhead or similar).

visible_printable (default value)

With this value all the visible layers as well as all the printable layers will be printed/exported.

visible

With this value only the visible layers will be printed/exported.

printable

With this value only the printable layers will be printed/exported.

all

With this value all the layers will be printed/exported., whether they are printable or not.

print_imageQuality (optional)

The optional parameter **print_imageQuality** defines whether the Result File in PDF or PS format should be created with a high or low image resolution. the available values are **highres** or **lowres**. If no setting is made, the default value **lowres** will be set.

highres

With this value, the original image file will be output. The value **highres** is generally used when the Result File is a PDF or PS document that is to be output to a high resolution device.

lowres

With this value, only the screen preview of the image file (72 DPI-image preview) is output. The value **lowres** is generally used when the output file is a PDF that is only used for correction/proofing purposes.

noimage

With this value, no images are output. The value **noimage** is generally used when the output file is a PDF that is only used for correction/proofing purposes.

Marks (optional)

The optional parameter **print_cropMarks** defines if the Result File should be output with crop marks. The accepted values are **true** or **false**.

The optional parameter **print_cropMarkLength** defines the length of the crop marks. The accepted value is a string.(e.g. »3 mm«).

The optional parameter **print_cropMarkDistance** defines the distance of the crop marks from the page.

The accepted value is a string.(e.g. »3 mm«).

The optional parameter `print_registrationMarks` defines if the Result File should be output with registration marks. The accepted values are **true** or **false**.

Bleed (optional)

The optional parameter `print_useBleed` defines if the Result File will be created with bleed. The accepted values are **true** or **false**.

The optional parameter `print_bleed` defines the size of the bleed. Alternatively, you can define a bleed for each edge of the page with the parameters `print_bleedLeft`, `print_bleedTop`, `print_bleedRight` and `print_bleedBottom` nieren. The accepted value is a string.(e.g. »3 mm«).

PDF Options(optional)

The optional parameter `print_pdfCompatibility` defines the format of the PDF file. the accepted values are **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**.

The optional parameter `print_pdfOptimize` defines if the PDF file should be optimized. The accepted values are **true** or **false**.

The optional parameter `print_pdfCompressionMethod` defines the compression for the PDF file. The accepted values are **none**, **ZIP** or **JPEG**.

The optional parameter `print_pdfCompressionQuality` defines the quality for the compression process. The accepted values are from **0** to **100** percent.

The optional parameter `print_pdfCompressText` defines if the text should be compressed in the PDF file. The accepted values are **true** or **false**.

The optional parameter `print_pdfWarningScale` defines if a warning for image scaling should be output in the PDF file. The accepted values are **true** or **false**.

The optional parameters `print_pdfWarningScaleBelow` and `print_pdfWarningScaleAbove` define the minimum and maximum image scale as a percentage (%). If the values are exceeded or not reached, a sticky note will be created containing the scaling values. The accepted values are from **0** to **100** percent. This parameter is only active when the parameter `print_pdfWarningScale` has the value **true**.

The optional parameter `print_pdfWarningTextOverflow` defines if a text overflow warning should be output in the PDF file. In this case, a sticky note will be created containing the first 200 characters of the overflow text. The accepted values are **true** or **false**.

ICC Profiles (optional)

The optional parameters `print_pdfxOutputIntent`, `print_pdfxGrayProfile`, `print_pdfxRGBProfile` and `print_pdfxCMYKProfile` define the ICC-Profile to be used. The accepted value is a string, that corresponds to the name of the profile (e.g. »CoatedFOGRA27.icc«).

The program gets the profiles from the folder »Resources/Color Profiles«, as profiles are NOT embedded in the native VIVA format. All profiles used must therefore be stored in this folder.

Various Parameters (optional)

The optional parameter **print_pageNumbering** defines the page numbering in the PDF file. The accepted values are **Physical**, **Logical** or **PhysicalAndLogical**.

The optional parameters **print_horizontalAlignment** and **print_verticalAlignment** define the document alignment. The application of this parameter is only useful when the spread or paper format saved in the document is larger than the document format (e.g. A5 format output on A4 paper format). The accepted values are **left**, **center** or **right** and **top**, **center** or **bottom**.

The optional parameter **print_fontEmbedding** defines if and how fonts should be embedded in the PDF file. The accepted values are **none**, **type1 but standard 14**, **type1** or **all**.

The optional parameter **print_printSpreads** definiert, ob nebeneinanderliegende Seiten als Bogen ausgegeben werden sollten. The accepted values are **true** or **false**.

The optional parameter **print_ignoreTrapping** defines if the automatic color trapping should be applied. The accepted values are **true** or **false**.

The optional parameter **print_printSpellCorrection** defines if spelling errors should be displayed in the Result File. The accepted values are **true** or **false**.

The optional parameter **print_opi** defines if images should be searched for on an OPI Server for the output. The accepted values are **none**, **TIFFOnly**, **EPSOnly**, **TIFFAndEPS** or **auto**.

colorSeparation (optional)

Der optionale Parameter **colorSeparation** defines if the Result file should be color-separated. For the color separation an individual color page will be created for every process and spot color used. The values **off**, **application** and **inrip** can be selected. If no entry is made, the default value **off** will be set. The use of this parameter only makes sense if the parameter **outType** has the value **pdf** or **ps**. If a color separation is required, we warmly recommend using the functions of a RIP, since here as well as a professional color separation (even of RGB images) even a high quality color trapping may be calculated.

off (default)

With this value no color separation is made. The Result file is made in "Composite Mode".

application

With this value, the color separation will be made directly by the VIVA Network Publishing Server. Here it must be noted that errors may occur if the PDF format (Parameter **outType**) is used for the Result file, and if the document created contains vector EPS or PDF files. If the Result file uses the PS format (Parameter **outType**), no problems will occur in this area.

inrip

With this value, the color separation is not made by the VIVA Network Publishing Server, but by a RIP. Irrelevant of the required format of the Result file (Parameter **outType**), the parameter delivers a PostScript file with the appropriate commands for the InRIP color separation. Please note the options of the RIPs used.

ZIP Parameter (optional)

The optional parameter **service_includeDocument** defines if in the creation of the ZIP file the document should be inserted. The accepted values are **true** or **false**.



The optional parameter **service_includeProtocol** defines if in the creation of the ZIP file the document statistics should be inserted. The accepted values are **true** or **false**.

The optional parameter **service_includeFonts** defines if in the creation of the ZIP file the should be inserted. The accepted values are **true** or **false**.

The optional parameter **service_includePictures** defines if in the creation of the ZIP file the images should be inserted. The accepted values are **true** or **false**.

The optional parameter **service_searchInSystemFontFolder** defines if in the creation of the ZIP file the fonts should be searched for in the system folder. The accepted values are **true** or **false**.

The optional parameter **service_searchInSpecialFontFolder** defines if in the creation of the ZIP file the fonts should be searched for in a special folder. The accepted values are **true** or **false**.

The optional parameter **service_specialFontFolder** defines the path for the special font folder. The accepted value is a string.

The optional parameter **service_useSubfolders** defines if for fonts and images the appropriate sub-folders should be created. The accepted values are **true** or **false**.



APPLICATION EXAMPLES

VIVA has set up a series of examples which create a business card from different applications. VIVA supplies these examples in source code and they serve the developer as a model for his own programs and for a better understanding of the technology.

Addressing via a programming language

1. PHP

You can steer a »VIVA Network Publishing Server« very comfortably via »PHP«. An extensive Demo application is an integral part of this documentation.

2. JAVA

You can steer a »VIVA Network Publishing Server« very comfortably via »JAVA«. An extensive Demo application is an integral part of this documentation.

3. BORLAND DELPHI

You can also steer a »VIVA Network Publishing Server« via »Borland Delphi«. An extensive Demo application is an integral part of this documentation.

Addressing via a database

1. Microsoft Access

You can also steer a »VIVA Network Publishing Server« via a »Microsoft Access« database. An extensive Demo application is in preparation and will shortly become an integral part of this documentation.

Addressing via an Internet Browser

You can also steer a »VIVA Network Publishing Server« directly via a URL, which is entered in an Internet Browser. The syntax for addressing a VIVA Network Publishing Server consists of three sections:

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

{VIVA NETWORK-PUBLISHING SERVER}

The Address of the VIVA Network Publishing Server

Example: <http://viva-nwp.de> or <http://196.168.0.20>

{SCRIPT}

Name of the Script that starts the request

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

{PARAMETER}

Parameters that are transferred to {VIVAIPSCRIPT}. The parameter list begins with a question mark (?). The syntax consists of the parameter name and the corresponding value. Any number of parameters may be entered, which are always separated from one another by an & (ampersand) character.

Syntax: ?{NAME1}={VALUE1}&{NAME2}={VALUE2}&{NAME3}=...

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

Below is an example for an application, which sends a request to the VIVA Network Publishing Server to create a document in PDF format consisting of pages two to five:

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

ADDRESSING VIA DOS COMMANDS

The syntax for addressing a VIVA Network Publishing Servers with DOS commands consists of three sections:

By contrast to Web/Application the process must run on the Server (or from where does the computer get the address??

{PROGRAM} {SCRIPT} {PARAMETER}

{PROGRAM}

Summon the PHP Program

Example c:\php4\php.exe

{SCRIPT}

Name of the Script that starts the request, whereby here the complete path must be entered. Only Slashes (/) and NO Backslash (\) may be used.

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

{PARAMETER}

Parameters which are transferred to {SCRIPT}. The parameter list begins with a (here & character??)question mark (?). The syntax consists of the parameter name and the corresponding value. Any number of parameters may be entered, which are always separated from one another by an & (ampersand) character.

Syntax: ?{NAME1}={VALUE1}&{NAME2}={VALUE2}&{NAME3}=...

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

{PARAMETER} Parameters that are transferred to "vivanwp.php" Script

Syntax: &{NAME1}={VALUE1}&{NAME2}={VALUE2}&{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"