



Server™

EULA, Copyright Notice & Trademarks

End User License Agreement

NOTICE TO USER:

THIS IS A CONTRACT. BY INSTALLING THIS SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT.

This activePDF, Inc. ("ACTIVEPDF") End User License Agreement ("EULA") accompanies all activePDF products and related explanatory materials ("SOFTWARE"). The term "SOFTWARE" shall also include any upgrades, modified versions or updates of the Software licensed to you by ACTIVEPDF. Please read this EULA carefully. If you are reading this prior to electronic distribution, you will be asked to accept this agreement and continue to install or, if you wish to decline this agreement, in which case you will not be able to use the Software.

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold.

1. DEFINITIONS

- I. SERVER - A single computer wholly owned, rented or leased by a single individual or entity on which one or more applications load and execute SOFTWARE in the memory space of that computer so that one or more users may access it.
- II. VIRTUAL SERVER - A single computer wholly owned, rented or leased by one individual or entity and who, in turn, rents or leases access to this computer to other individuals or entities and on which one or more applications load and execute SOFTWARE in the memory space of that computer so that multiple users may access it.
- III. DEVELOPMENT - The act of programming a tool or application that interacts with SOFTWARE.
- IV. APPLICATION SERVICE PROVIDER - An individual or entity ("OWNER") and who, in turn, charges other individuals or entities a fee for access to software applications wholly owned or licensed and maintained by the OWNER. The OWNER's application(s) load and execute SOFTWARE in the memory space of that computer so that multiple users may access it.

2. GRANT OF LICENSE : Upon issuance of a valid SERIAL NUMBER (for SERVER and VIRTUAL SERVER LICENSES) by ACTIVEPDF to you, this EULA grants you the following rights:

a. SOFTWARE PRODUCT : ACTIVEPDF grants to you as an individual, a non-transferable, nonexclusive license to use the SOFTWARE PRODUCT as follows:

I. IF YOU PURCHASED A SERVER LICENSE - Usage is restricted to a single SERVER for a single individual or entity. All SERVERS that use this product must be licensed. III. IF YOU PURCHASED A VIRTUAL SERVER LICENSE - Usage is restricted to a single individual or entity. All other individuals or entities that can access or use SOFTWARE must purchase a separate VIRTUAL SERVER LICENSE.

II. If you Purchased a Development License or a Developer Bundle. (SALES AND GRANT OF DEVELOPER LICENSES ARE AT ACTIVEPDF'S DISCRETION AND ALWAYS REQUIRE THE PURCHASE OR EXISTENCE OF VALID PRODUCTION LICENSES) You are hereby granted the right for development use only for the license or licenses purchased as development licenses. You must purchase a license for each developer granted access and for each development machine where you load the software. Development software is for Development purposes only and you may not use Development licenses in a live test or production environment (Test Servers require full licenses not Development Licenses). You may not resell, transfer, rent, or lease the SOFTWARE. You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble, or otherwise attempt to discover the source code of the Software. You may not alter or modify in any way the installer for the Software, or create a new installer for the Software. You also agree to hold in the strictest confidentiality any and all code given to you by ACTIVEPDF to enable your application.

b. If you are an entity, ACTIVEPDF grants you the right the same rights as above.

c. UNTIL A VALID SERIAL NUMBER HAS BEEN ISSUED TO YOU BY ACTIVEPDF, YOU MAY ONLY USE THE SOFTWARE FOR A TRIAL PERIOD NOT TO EXCEED FIFTEEN (15) DAYS FROM THE INITIAL INSTALLATION. YOU AGREE TO REMOVE ANY COPIES OF THE SOFTWARE UPON EXPIRATION OF THE TRIAL PERIOD IF YOU DO NOT PURCHASE THE SOFTWARE. YOU MAY NOT DISTRIBUTE ANY PORTION OF THE SOFTWARE, INCLUDING THOSE PORTIONS OUTLINED UNDER "REDISTRIBUTABLE CODE" BELOW, UNTIL YOU ARE ISSUED A VALID SERIAL NUMBER. NO LICENSE IS GRANTED UNTIL THAT TIME.

d. ELECTRONIC DOCUMENTS : Solely with respect to electronic documents included with the SOFTWARE, you may make an unlimited number of copies (either in hardcopy or electronic form), provided that such copies shall be used only for internal purposes and are not republished or distributed to any third party.

e. STORAGE/NETWORK USE : You may store or install one (1) copy of the SOFTWARE on a storage device, such as a network server, for backup and archival purposes only. A license for the SOFTWARE may not be shared or used concurrently on different computers.

f. You agree not to modify the Producer or Creator fields within any PDF documents created by SOFTWARE.

g. ACTIVEPDF reserves all rights not expressly granted.

3. COPYRIGHT. The SOFTWARE is owned by ACTIVEPDF and its suppliers, and its structure, organization and code are the valuable trade secrets of ACTIVEPDF and its suppliers. The SOFTWARE is protected by United States Copyright Law and International Treaty provisions. You may use trademarks only insofar as required to comply with Section 1 of this EULA and to identify printed output produced by the Software, in accordance with accepted trademark practice, including identification of trademark owner's name. Such use of any trademark does not give you any rights of ownership in that trademark. Except as stated above, this EULA does not grant you any intellectual property rights in the Software.

4. RESTRICTIONS. You may not resell, transfer, rent or lease the SOFTWARE. You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software. You may not alter or modify in any way the installer for the Software, or create a new installer for the Software. Redistribution rights, if any, are outlined below.

5. ADDITIONAL RESTRICTIONS.

I. For activePDF Server, activePDF DocConverter or activePDF WebGrabber : You agree that you will only use SOFTWARE on a SERVER and all applications that will access SOFTWARE will reside on the SERVER and you will not permit remote access to SOFTWARE except through your application residing on the SERVER. Additionally, you agree not to use SOFTWARE in a generic, bureau-like fashion whereby the only functionality of your application is to submit files for conversion to PDF.

II. For activePDF Server : You agree to not attempt to violate the internal licensing mechanism for processing thread increases. You also agree to not attempt to defeat said internal licensing mechanism by, directly or indirectly, halting and restarting the SOFTWARE.

III. For activePDF DocConverter : You agree that you will not charge any fees, directly or indirectly, internally or externally, for access to SOFTWARE. This restriction includes charging per document, for the right to convert documents on your server and/or as a requirement of any sort. You also agree that should you violate this restriction, you will pay ACTIVEPDF an amount commensurate to the total amount collected by you for said restricted services or \$2,000 US, whichever is more. You also agree to surrender your license(s) immediately upon notification of violation of this restriction and that you will not be refunded any monies upon surrendering your license. You agree to not attempt to violate the internal licensing mechanism for document processing increases. You also agree to not attempt to defeat said internal licensing mechanism by, directly or indirectly, halting and restarting the SOFTWARE.

IV. For activePDF WebGrabber : You agree to only render HTML to which you have a legal and lawful right to access and that you will not violate any individual's or entity's copyright protection with unauthorized access to that individual's or entity's content. You also agree to indemnify and hold harmless ACTIVEPDF from any such activity even if such violation is unintentional.

V. For activePDF DocConverter : Installation and use by an APPLICATION SERVICE PROVIDER requires a separate, signed agreement and is prohibited by this EULA.

6. UPGRADES. If the SOFTWARE is labeled as an upgrade, you must be properly licensed to use a product identified by ACTIVEPDF as being eligible for the upgrade in order to use the SOFTWARE. SOFTWARE labeled as an upgrade replaces and/or supplements the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this EULA. If the SOFTWARE is an upgrade of a component of a package of software programs that you licensed as a single product, the SOFTWARE may be used and transferred only as part of that single product package and may not be separated for use on more than one computer. Development licenses are not eligible for upgrades or other rights granted to products purchased with subscription.

7. REDISTRIBUTABLE CODE :

I. SERVER LICENSE - No portion of SOFTWARE PRODUCTS may be redistributed.

II. VIRTUAL SERVER LICENSE - No portion of SOFTWARE PRODUCTS may be redistributed.

8. EXPORT RESTRICTIONS. You agree that neither you nor your customers intend to or will, directly or indirectly, export or transmit (a) the SOFTWARE or related documentation and technical data or (b) your software products as defined under of this EULA (or any part thereof), or any process or service that is the direct product of the SOFTWARE to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce, or such other governmental entity as may have jurisdiction over such export or transmission.

9. NO WARRANTY. The SOFTWARE is being delivered to you AS IS and ACTIVEPDF makes no warranty as to its use or performance. ACTIVEPDF AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. ACTIVEPDF AND ITS SUPPLIERS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL ACTIVEPDF OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, EVEN IF AN ACTIVEPDF REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY THIRD PARTY. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you.

10. GOVERNING LAW AND GENERAL PROVISIONS. This EULA will be governed by the laws of the State of California, U.S.A., excluding the application of its conflicts of law rules. This EULA will not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. If any part of this EULA is

found void and unenforceable, it will not affect the validity of the balance of the EULA, which shall remain valid and enforceable according to its terms. You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations. This EULA shall automatically terminate upon failure by you to comply with its terms. This Agreement may only be modified in writing signed by an authorized officer of ACTIVEPDF.

11. Notice to Government End Users. If this product is acquired under the terms of a: GSA contract- Use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract; U.S. DoD contract- Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of 252.227-7013; Civilian agency contract- Use, reproduction, or disclosure is subject to 52.227-19 (a) through (d) and restrictions set forth in this EULA.

Unpublished-rights reserved under the copyright laws of the United States. activePDF, Inc., a California Corporation, 27405 Puerta Real, Suite 100, Mission Viejo, California 92691.
Effective 3/1/2005

Copyright Notice

Copyright © 2000-2006, activePDF, Inc., All Rights Reserved
Portions of this publication copyright © 2000 Glance AG. All Right Reserved.
Released 02/2005 by activePDF, Inc.

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent in writing from activePDF, Inc. 27405 Puerta Real, Suite 100, Mission Viejo, CA 92691-6314.

ALL EXAMPLES OF NAMES OR COMPANIES APPEARING IN THIS MANUAL ARE FICTIONAL. ANY SIMILARITY TO ACTUAL NAMES OR COMPANIES IS PURELY COINCIDENTAL.

Every effort has been made to ensure accuracy of this manual. However, activePDF, Inc. makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. activePDF, Inc. shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance or use of this manual or the examples herein. The information in this document is subject to change without notice.

Trademarks

activePDF®, the activePDF Logo, activePDF Server™, and activePDF DocConverter™ are all registered trademarks or trademarks of activePDF, Inc. respectively. Adobe®, Adobe Acrobat® and Adobe Acrobat Reader® are registered trademarks of Adobe® Systems, Inc. Microsoft®, Microsoft Windows®, Windows NT® and Windows 2000® are registered trademarks of Microsoft Corporation. All product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are the sole property of their respective manufacturers.

Dedication

Dedicated to the memory of Arleigh Rogers (1979-2006).

Contacting activePDF, Inc.

- **activePDF, Inc. Corporate Headquarters**
- **Technical Support**
- **Sales**

activePDF, Inc. Corporate Headquarters

activePDF is a leading provider of PDF generation and manipulation tools. Our products are easy to use, enabling you to create PDF files that are compatible with 3rd party PDF tools and conform to the latest version of the PDF Specification, published by Adobe® Systems. **activePDF, Inc.** 27405 Puerto Real Suite 100 Mission Viejo, California 92691-6314 United States *Toll Free (U.S.): (866) GOTO PDF Elsewhere: +1-949-582-9002 Fax: (949) 582-9004 World Wide Web:*
www.activepdf.com

Technical Support

At activePDF, we are committed to providing you with timely answers to all of your technical support questions. For current hours of operation and details about support offerings, please visit the activePDF website at <http://www.activepdf.com/support/supportprogram/supportservices/>. activePDF's support services are subject to the terms and conditions in place at the time the service is used.

Sales

activePDF is a leading provider of server-side PDF generation and manipulation tools. For information on other activePDF product offerings, please contact activePDF's Sales Department by calling (866) GOTO PDF in the U.S., or +1-949-582-9002 elsewhere. You can also visit us on the web at www.activepdf.com.

Table of Contents

About this Document	12
Getting Started	13
System Requirements	14
Installation	15
activePDF License Administrator	16
Installing the Software	18
Upgrading Server's Version or Edition	20
Upgrading from a Previous Version or Service Pack	21
Upgrading from an Evaluation Edition	22
Increasing the Number of Threads	23
Testing Your Installation	25
Server Configuration Editor	26
General Settings	27
Color Image Settings	28
Gray Image Settings	31
Monochrome Image Settings	33
Output Settings	35
Advanced Settings	37
Configuration Options	39
Terminal Server Settings	40
Default Printer	42
Printer & Pool Types	43
System Event Log	44
Object Reference	45
Instantiating the Object	46
Methods	47
AbortPrinting	48
AddFileBookmark	49
AddLinkedPDFBookmark	50
AddPageBookmark	51
AddPDFMark	52
AddURLBookmark	53
ClearPDFMarks	54
DeleteFile	55
FromString	56
GetBinaryImage	57
GetExclusiveAccess	58
ImageToPDF	59
ImpersonateUser	61
IsFingerprintValid	62
LoadConfiguration	63
LoadImageInfo	64
LoadRemoteProfile	65
PDFToBrowser	66
PrintToBrowser	67
PSToPDF	68
SetColorDownsampleThreshold	69
SetGrayDownsampleThreshold	70
SetMonoDownsampleThreshold	71

SetOutputSecurity	72
SetOutputSecurity128	74
SetPrinterAsDefault	76
SetViewMode	77
StartPrinting	78
StopImpersonation	79
StopPrinting	80
Test	81
ToString	82
Wait	83
XMLData	84
Properties	85
ASCIIEncode	87
AutoRotate	88
ChunkSize	89
ColorDownsampleType	90
ColorFilter	91
ColorImageFlags	92
ColorImageResolution	94
CompatabilityLevel	95
ConcatenateToExisting	96
DefaultDomain	97
DefaultPassword	98
DefaultPrinter	99
DefaultQueue	100
DefaultUser	101
DocumentName	102
EmbedAllFonts	103
ExtendedError	104
FingerprintPDF	105
FlateCompression	106
FontSubsetting	107
FormName	108
FullDocumentPath	109
GeneralFlags	110
GraphicBitsPerPixel	112
GraphicsQuality	113
GrayDownsampleType	114
GrayFilter	115
GreyImageFlags	116
GreyImageResolution	118
ImageInfoBitsPerPixel	119
ImageInfoFileType	120
ImageInfoHeight	124
ImageInfoTotalPages	125
ImageInfoWidth	126
ImageInfoXResolution	127
ImageInfoYResolution	128
KeepPortDocument	129
LastError	130
LinearizeDocument	131
MonoFilter	132
MonoImageFlags	133
MonoImageResolution	134
NewDocumentName	135

NewPortName	136
NewPrinterName	137
NewUniqueID	138
Orientation	139
OutputDirectory	140
PaperLength	141
PaperSize	142
PaperWidth	144
PDFAuthor	145
PDFKeywords	146
PDFSubject	147
PDFTimeout	148
PDFTitle	149
PoolServerPort	150
PreserveHalftone	151
PreserveOverprint	152
PrinterPoolWaitTimeout	153
PrintQuality	154
PrintResolution	155
ReservedPrinterNumber	156
Resolution	157
Scaling	158
TrueTypeOption	159
UCRandBGR	160
UseStaticPool	161
Advanced Configuration	162
Using Server in .NET	163
Using Registry Switches	165
activePDF FileCleaner Utility	167
Appendices	169
PDF Coordinates and Units	170
PDF Views	171
PDFMarks	175
Development Examples	176
Word - VBScript	177
Word - ASP	179
Visual FoxPro	181
Clarion	182
Delphi	183
Server Updates	184

About this Document

This document provides information for deploying activePDF Server in a development environment. It is organized into the following sections:

- ▶ **Getting Started** - Covers the System Requirements, Installation, Testing your Installation, and explains How Server Generates a PDF.
- ▶ **Configuration Editor** - Details the activePDF Server Configuration Editor program options.
- ▶ **Advance Setup** - Contains additional setup requirements for Fonts, the Default Printer, explains the Printer and Pool Types, and Event Logging.
- ▶ **Object Reference** - Includes object instantiation instructions and lists all methods and properties.
- ▶ **Advanced Configuration** - Covers the use of Server in .NET, Remote Connection Environments, details the Quality Settings, Font Embedding, Registry Switches, and the File Cleaner Utility.
- ▶ **Appendices** - Provides an explanation of PDF Units, details PDF Views and Marks, contains Development Environment Examples, and the Update History.

Who Should Read This Guide

This guide has been written for the developer who wants to programmatically generate PDFs and control the resultant output. The guide assumes you have a general knowledge of PostScript® and PDF, and that you are comfortable programming in a COM-enabled environment.

Documentation Feedback

activePDF strives to produce quality technical documentation. If you have comments or suggestions regarding our help files, PDF, print or online manuals please send an email to the [activePDF Documentation Team](#). Please include the following information in your message:

- ▶ Product name and version number
- ▶ Print manual, PDF or help file
- ▶ Section or Topic title
- ▶ Brief description of content
- ▶ Your suggestion for improvement or correction

NOTE: This email address is only for documentation feedback. If you have a technical question, please contact [Technical Support](#).

Getting Started

activePDF Server is a robust, server-side application that generates Portable Document Format (PDF) files using a COM interface, including PostScript®-to-PDF and Image-to-PDF conversions. Whether converting large batches of documents or handling individual conversions on a job-by-job basis, activePDF Server makes it easy to customize your PDF output. Server makes it easy to add PDF conversion functionality to your existing enterprise applications, and provides multi-threaded functionality, making it ideal for high-volume situations. It runs *Out-of-Process* on your server-machine, which means there is no need for interaction or monitoring.

- ▶ **System Requirements**
- ▶ **Installation**

System Requirements

Operating System Requirements

- ▶ Microsoft® Windows NT® 4.0 (Service Pack 5 minimum) or
- ▶ Microsoft Windows® 2000 or
- ▶ Microsoft Windows Server 2003
- ▶ Strong Encryption

NOTE: Strong encryption is only required for 128-bit encryption.

Minimum Recommended Hardware Requirements

- ▶ Pentium III 500-MHz or higher
- ▶ 128 MB of RAM
- ▶ 15MB of Hard Disk Space (for application)
- ▶ 100 MB of Hard Disk Space (for spooling, depending upon application)

Port Requirements

- ▶ By default, port 53535 must remain open and dedicated for Server.

Installation

Installing activePDF Server requires an *Activation Key*, the [activePDF License Administrator](#), and the installation software. The *Activation Key* is supplied via email upon purchase. If you are installing the evaluation edition, you will require an Evaluation Activation Key which can be obtained by contacting the [activePDF Sales Team](#).

- ▶ [activePDF License Administrator](#)
- ▶ [Installing the Software](#)
- ▶ [Upgrading Server's Version or Edition](#)
- ▶ [Increasing the Number of Threads](#)
- ▶ [Testing Your Installation](#)

```
</PLAINTEXT><PRE></PRE></Q></S></SAMP></SCRIPT></SELECT></SMALL></STRIKE></STRONG></SUB></
<UL></UL></VAR></WBR><XMP></XMP><FONT face=arial></FONT>
<SCRIPT language=JavaScript>
  function showHide(targetName) {
    if( document.getElementById ) { // NS6+
      target = document.getElementById(targetName);
    } else if( document.all ) { // IE4+
      target = document.all[targetName];
    }

    if( target ) {
      if( target.style.display == "none" ) {
        target.style.display = "inline";
      } else {
        target.style.display = "none";
      }
    }
  }
</SCRIPT>
```

```
<TABLE borderColor=#000808 cellPadding=3 bgColor=#e7e7e7 border=1>
<TBODY>
<TR>
<TD bgColor=#000066><FONT style="FONT: 11pt/13pt verdana; COLOR: white"
  color=white>Error Occurred While Processing Request</FONT></TD></TR>
<TR>
<TD><FONT style="FONT: 8pt/11pt verdana; COLOR: black">
<TABLE cellSpacing=0 cellPadding=0 width=500 border=0>
<TBODY>
<TR>
<TD id=tableProps2 vAlign=center align=left width=500>
<H1 id=textSection1
  style="FONT: 13pt/15pt verdana; COLOR: black">Variable VARPRODUCT is
  undefined. </H1></TD></TR>
<TR>
<TD id=tablePropsWidth width=400 colSpan=2><FONT
  style="FONT: 8pt/11pt verdana; COLOR: black"></FONT></TD></TR>
<TR>
<TD>&nbsp;</TD></TR>
<TR>
<TD colSpan=2><FONT
  style="FONT: 8pt/11pt verdana; COLOR: black">Please try the
  following:
<UL>
<LI>Enable Robust Exception Information to provide greater detail
  about the source of errors. In the Administrator, click Debugging
  & Logging & Debugging Settings, and select the Robust
  Exception Information option.
<LI>Check the <A
  href="http://www.macromedia.com/go/proddoc_getdoc"
  target=new>ColdFusion documentation</A> to verify that you are
  using the correct syntax.
<LI>Search the <A
  href="http://www.macromedia.com/support/coldfusion/"
  target=new>Knowledge Base</A> to find a solution to your problem.
</LI></UL>
<P></P></FONT></TD></TR>
<TR>
<TD colSpan=2>
<TABLE cellSpacing=0 cellPadding=0 border=0>
<TBODY>
```

```
<TR>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Browser&nbsp;&nbsp;&nbsp;</FONT><
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Mozilla/4.0
    (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; .NET
    CLR 2.0.50727)</FONT></TD></TR>
<TR>
  <TD><FONT style="FONT: 8pt/11pt verdana; COLOR: black">Remote
    Address&nbsp;&nbsp;&nbsp;</FONT></TD>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">216.23.175.190</FONT></TD><
<TR>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Referrer&nbsp;&nbsp;&nbsp;</FONT>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black"></FONT></TD></TR>
<TR>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">Date/Time&nbsp;&nbsp;&nbsp;</FONT>
  <TD><FONT
    style="FONT: 8pt/11pt verdana; COLOR: black">16-Oct-07 03:11
    PM</FONT></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></FONT></TD
```

Installing the Software

Production and Evaluation Editions of activePDF Server are available for download from <http://www.activepdf.com/downloads/serverproducts/index.cfm>.

NOTE: The activePDF Server Service must run under the *Local System* account.

1. Log in as the **Local Administrator**
2. Insert your activePDF CD or download activePDF Server from <http://www.activepdf.com/downloads/serverproducts/index.cfm>.
3. Double-click **APServer.EXE** and follow the onscreen prompts to install the software files.
4. Once the software files are installed, you will need to enter your *Activation Key*.

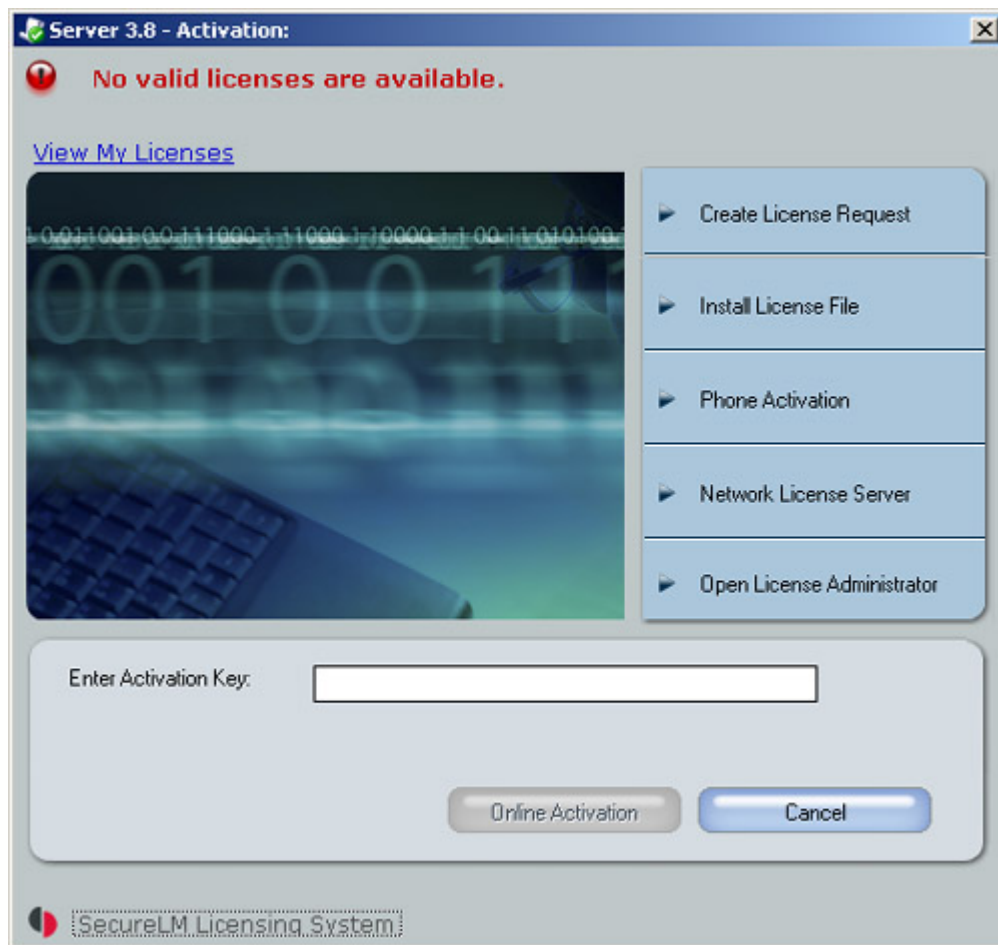


Figure 2.2: The activePDF Software Activation Screen.

The *Activation Key* is supplied via email upon purchase. If you are installing the evaluation edition, you will require an *Evaluation Activation Key* which can be obtained by contacting the [activePDF Sales Team](#).

5. After entering your *Activation Key*, click **Online Activation** to activate your license. A message box will display indicating successful activation.

NOTE: Online Activation requires an internet connection. If you are not connected to the internet, please contact the

activePDF Sales Team to activate by phone.

Upgrading Server's Version or Edition

The [activePDF License Administrator](#) simplifies the process of upgrading from an evaluation version to a production version, or from a previous version of the software.

- › [Upgrading from a Previous Version or Service Pack](#)
- › [Upgrading from an Evaluation Edition](#)

Upgrading from a Previous Version or Service Pack

You must remove any prior installations of activePDF Server before installing version 3.8. To upgrade to Server 3.8 from a previous version:

1. Click **Start**, point to **Settings**, click **Control Panel**, and then double-click **Add/Remove Programs**.
 2. Locate and click **activePDF Server** in the list of installed programs, and then click **Add/Remove**.
 3. Click **Remove activePDF Server** and then click **OK**.
 4. Follow the instructions in [Installing the Software](#) to complete the installation.
-

NOTE: In addition, you must update your [.NET Wrapper](#).

Upgrading from an Evaluation Edition

The evaluation edition of activePDF Server is a fully functional and free to use. The evaluation period expires after 15 days and all output contains an activePDF watermark. After purchasing a production license for activePDF Server, you can upgrade your evaluation edition online through the [activePDF License Administrator](#). To upgrade from an evaluation version:

1. Choose **Start>Programs>activePDF>License Administrator**
2. Click **Activation**, then click **Re-Activate**. The Reactivating screen appears and will indicate success upon completion.

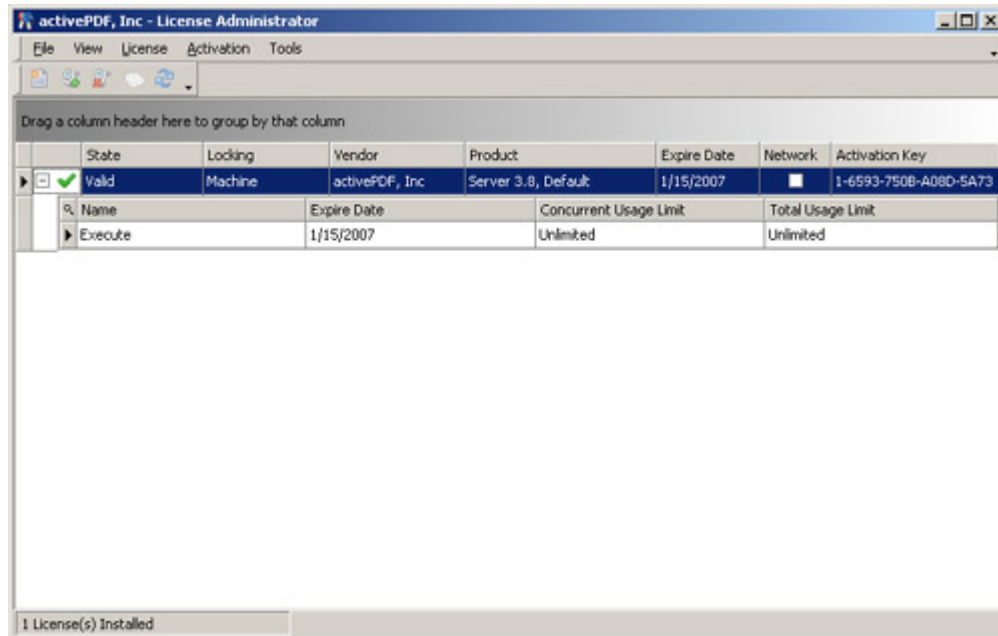


Figure 2.3: The Reactivating screen.

NOTE: Online Activation requires an internet connection. If you are not connected to the internet, please contact the [activePDF Sales Team](#) to activate by phone.

Increasing the Number of Threads

The production edition of Server includes support for up to two simultaneous threads. Increases are available for purchase in three thread increments and can be added using the [activePDF License Administrator](#). For further information or to purchase additional thread licenses, contact [activePDF Sales](#).

NOTE: Multiple processors are required for true simultaneous conversion. We strongly recommend that you consult [Technical Support](#) prior to purchasing to ensure your deployment will benefit from additional thread licenses.

To add additional thread licenses:

1. Choose **Start>Programs>activePDF>License Administrator**
2. Click **Activation**, then click **Re-Activate**. The Reactivating screen appears and will indicate success upon completion.

NOTE: Online Activation requires an internet connection. If you are not connected to the internet, please contact the [activePDF Sales Team](#) to activate by phone.

You can verify the additional licenses through the License Administrator by clicking the plus sign (+) to expand the activePDF Server license and display the additional threads.

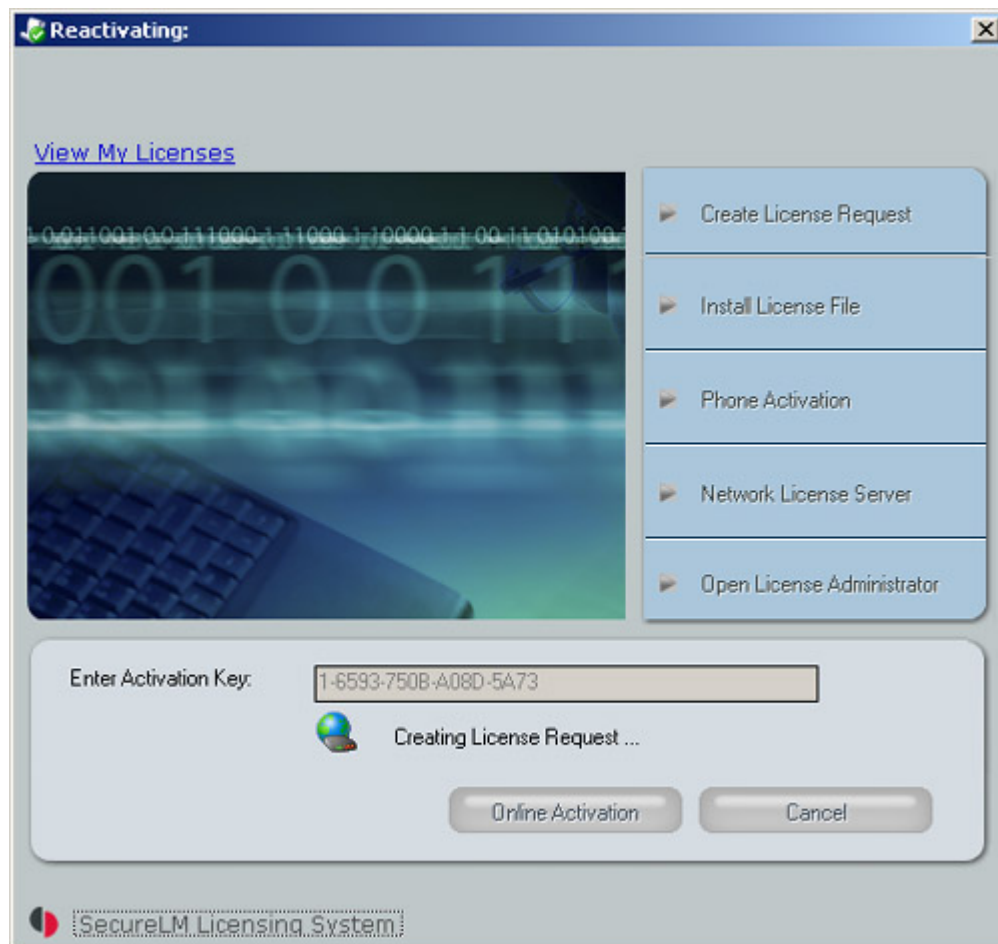


Figure 2.4: The activePDF License Administrator.

Testing Your Installation

The *activePDF Server Test Utility* installs with activePDF Server and is designed to provide a simple test to verify your installation. To run the utility:

1. Log on as the **Local Administrator**.
 2. Click **Start > Programs > activePDF > Server > Test Utility**.
-

The Test Utility will initiate a command window and then provide the results of the test. If the Test Utility does not return a success, contact [activePDF Technical Support](#).

Server Configuration Editor

The Configuration Editor enables you to apply settings to be used by activePDF Server in the PDF generation process. These settings control various PDF file characteristics, such as image compression and font embedding. In addition, the Configuration Editor includes options for controlling activePDF Server operation and performance, such as printer emulation and processing options.

NOTE: By default, settings in the Configuration Editor are applied to all PDFs generated through activePDF Server. Settings set via the API take precedence over corresponding settings in the Configuration Editor. For example, if set programmatically, the AutoRotate API property supersedes the value set by the Auto-Rotate option in the General Options of the Configuration Editor.

To start the activePDF Server Configuration Editor:

1. Log in as the **Local Administrator**.
2. Click **Start**, point to **Programs > activePDF > Server**, and then click **Configuration Editor**.

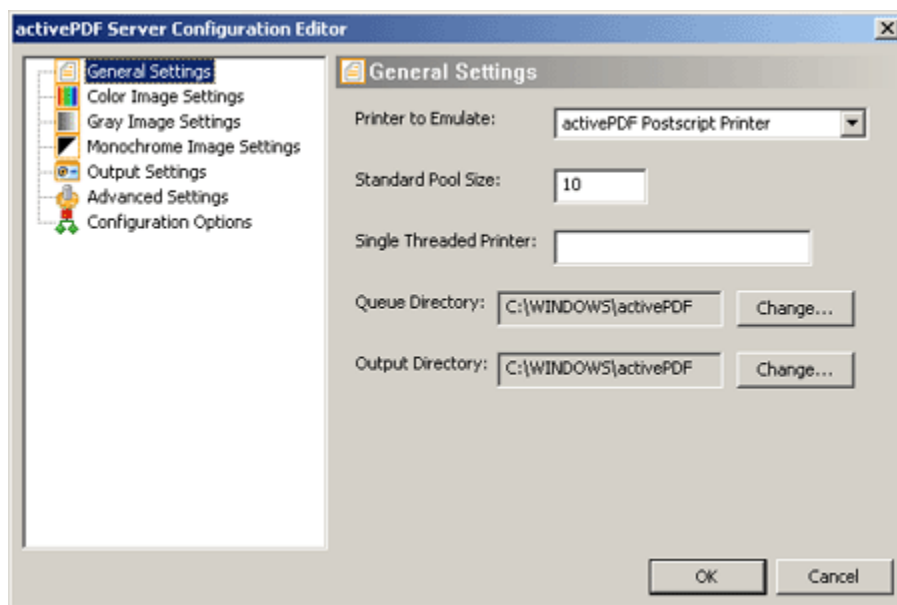


Figure 3.1: The Configuration Editor Dialog.

- ▶ General Settings
- ▶ Color Image Settings
- ▶ Gray Image Settings
- ▶ Monochrome Image Settings
- ▶ Output Settings
- ▶ Advanced Settings
- ▶ Configuration Options
- ▶ Terminal Server Settings

General Settings

The General Settings pane enables you to specify general operation settings for activePDF Server such as the default printer to emulate and the queue and output paths.

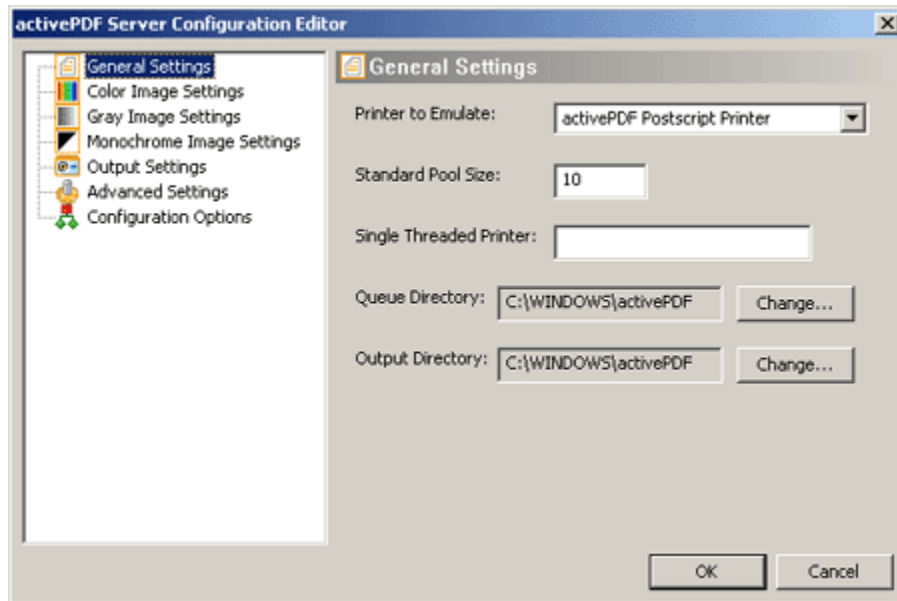


Figure 3.2: General Settings

- ▶ **Printer to Emulate:** The PostScript printer activePDF Server emulates when creating printers for print processing. Refer to [Printer & Pool Types](#) for additional information.
 - ▶ **Standard Pool Size:** The maximum number of dynamically-created printers (defined by the **Printer to Emulate** setting) that can exist concurrently in the Dynamic Pool. By default, this value is set to 10. Refer to [Printer & Pool Types](#) for additional information.
 - ▶ **Single Threaded Printer:** Enables you to specify a name for the single-threaded printer to be created by activePDF Server for the next print request, based on the printer specified in **Printer to Emulate**. Single-threaded printing disables the creation of dynamic printers, which can help control the print flow by forcing all jobs to go through a single printer. This setting might be used if you need to decrease the number of printers or ports used, or to increase stability.
- NOTE:** All subsequent print requests will be processed through this printer until removed. Leaving this setting blank will cause activePDF Server to continue generating a dynamic printer pool.
- ▶ **Queue Directory:** The temporary spooling location used by activePDF Server during the PDF generation process. Defaults to the root Windows directory.
 - ▶ **Output Directory:** The location for outputted PDFs. Defaults to the root Windows directory.

Color Image Settings

The Color Image Settings pane includes compression and resampling options for color images, enabling you to optimize your PDF to achieve the best balance between visual quality and resultant file size.

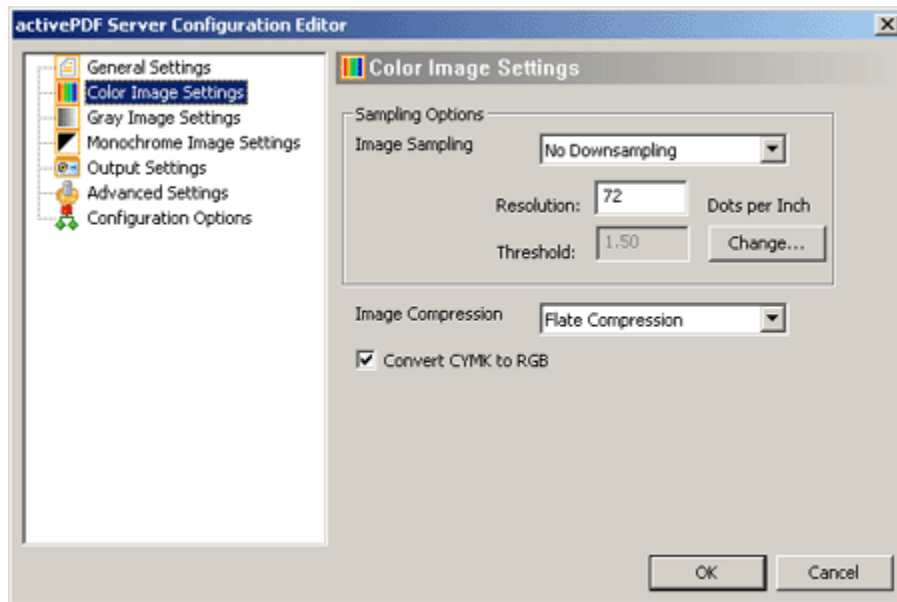


Figure 3.3: Color Image Settings

- Sampling Options:** Sampling options specify downsampling or subsampling for color images, using the specified resolution parameters. By altering the settings, you can minimize the loss of detail while maximizing reduction in file size. For example, if you are generating a PDF that is intended for onscreen display, you may wish to downsample all images to 72 DPI to reduce file size as well as the time it takes to display the image.

Image Sampling	Downsampling or subsampling reduces the number of dots per inch (DPI) in an image by combining dots in a sample area to create one larger dot. Since there is less information that needs to be saved, the resultant PDF file sizes can be considerably smaller. Sampling options include:
No Downsampling	No color images are downsampled in the output PDF file. This option produces the highest quality of output with no impact to file size, and might be used if you are creating PDF files for printing to a commercial-quality printer.
Average Downsampling	Averages neighboring dots, replacing dots of similar value with the average dot color. Averaging generally produces good quality results and is suitable for printing to most desktop printers, when set to the proper resolution.
Bicubic	Downsampling occurs using bicubic interpolation to generate new dot values at the specified resolution. In plain language, bicubic interpolation averages the dots in a sample group and replaces the entire group with an average of the dot color at the specified resolution. Bicubic downsampling is more precise, resulting in smoother gradations than other subsampling methods, although the PDF file is slower to generate.
Subsample	Subsampling takes a dot from the center of a sample group and replaces the entire

group with the selected dot at the specified resolution. Subsampling results in smaller file sizes and faster PDF creation, but you may notice a decrease in visual quality.

Resolution The target resolution (measured in dpi) for downsampling color images using the selected Image Sampling method.

Threshold The threshold ratio for downsampling images to the resolution specified. For example, if resolution is set to 300 and threshold is set to 1.5, images in your input file with a resolution above 450 dpi will be downsampled to 300 dpi ($300 * 1.5 = 450$). Click **Change** to calculate the downsampling threshold.

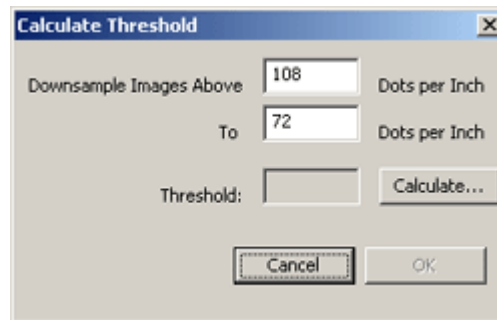


Figure 3.3: Calculate Threshold

Downsample Images Above # Dots per Inch To # Dots per Inch Downsampling is applied to color images with resolution exceeding the specified dots per inch (dpi), reducing the number of dpi to the specified resolution, using the selected Image Sampling method.

NOTE: If the No Downsampling option is selected, the Resolution and Threshold settings do not result in any change in the output file.

-
- ▶ **Image Compression:** There are two types of image compression: lossless and lossy. Lossless compression reduces the file size of an image file, without loss of data, by removing only repetitive information. Lossy compression also reduces file size, but by removing data from the image. When an image is compressed using lossy compression, its visual quality can differ from the original, particularly at higher magnification levels. Because lossy compression discards data, it can achieve smaller file sizes than lossless compression. Compression options include:

No Compression No compression is applied.

JPEG Compression The JPEG (Joint Photographic Experts Group) filter uses a lossy compression technique, encoding images as JPEGs to achieve a smaller file size. This method is typically recommended for continuous-tone photographic images which have more detail than can be reproduced in the desired output device (e.g. monitor or printer).

Flate Compression The Flate filter uses a lossless compression technique and typically produces good compression ratios. Flate is most useful in compressing images comprised of large areas of single colors or patterns, such as screen shots. Flate is also known as ZIP compression.

-
- ▶ **Convert CMYK to RGB:** When selected, converts CMYK colors to RGB. The CMYK color model is used for printing, while RGB is used for onscreen display.

Gray Image Settings

The Gray Image Settings pane includes compression and resampling options for grayscale images, enabling you to optimize your PDF to achieve the best balance between visual quality and resultant file size.

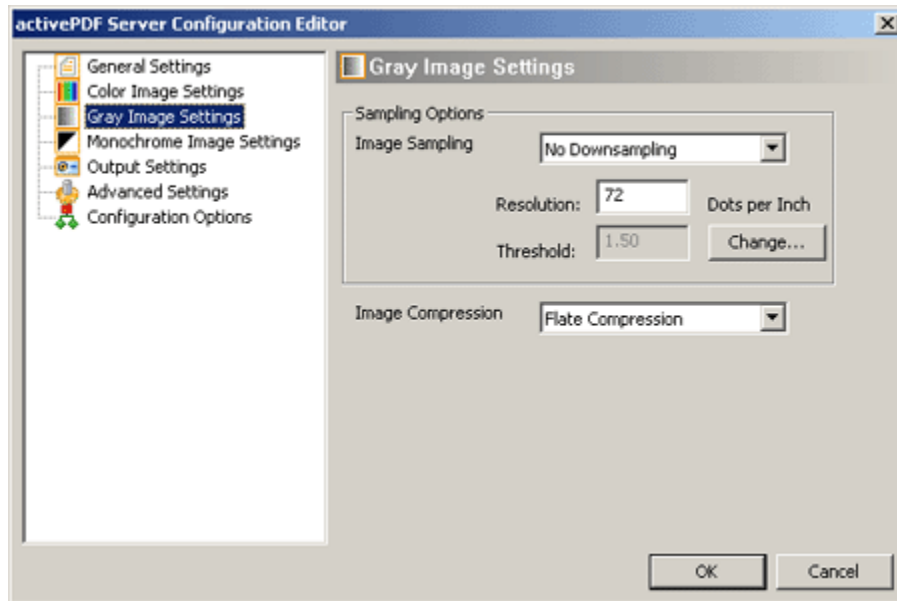


Figure 3.4: Gray Image Settings

- Sampling Options:** Sampling options specify downsampling or subsampling for grayscale images, using the specified resolution parameters. By altering the settings, you can minimize the loss of detail while maximizing reduction in file size. For example, if you are generating a PDF that is intended for onscreen display, you may wish to downsample all images to 72 DPI to reduce file size as well as the time it takes to display the image.

Image Sampling

Downsampling or subsampling reduces the number of dots per inch (DPI) in an image by combining dots in a sample area to create one larger dot. Since there is less information that needs to be saved, the resultant PDF file sizes can be considerably smaller. Sampling options include:

- No Downsampling** No grayscale images are downsampled in the output PDF file. This option produces the highest quality of output with no impact to file size, and might be used if you are creating PDF files for printing to a commercial-quality printer.
- Average Downsampling** Averages neighboring dots, replacing dots of similar value with the average dot color. Averaging generally produces good quality results and is suitable for printing to most desktop printers, when set to the proper resolution.
- Bicubic** Downsampling occurs using bicubic interpolation to generate new dot values at the specified resolution. In plain language, bicubic interpolation averages the dots in a sample group and replaces the entire group with an average of the dot color at the specified resolution. Bicubic downsampling is more precise, resulting in smoother gradations than other subsampling methods, although the PDF file is slower to generate.
- Subsample** Subsampling takes a dot from the center of a sample group and replaces the entire group with the selected dot at the specified resolution. Subsampling results in

smaller file sizes and faster PDF creation, but you may notice a decrease in visual quality.

Resolution The target resolution (measured in dpi) for downsampling grayscale images using the selected Image Sampling method.

Threshold The threshold ratio for downsampling images to the resolution specified. For example, if resolution is set to 300 and threshold is set to 1.5, images in your input file with a resolution above 450 dpi will be downsampled to 300 dpi ($300 * 1.5 = 450$). Click **Change** to calculate the downsampling threshold.

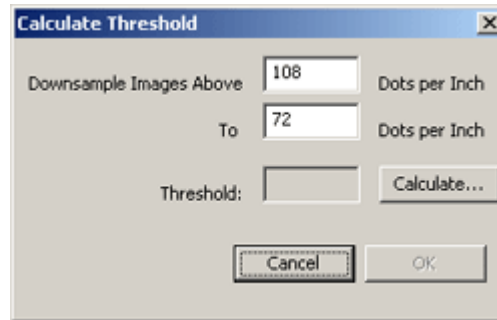


Figure 3.3: Calculate Threshold

Downsample Images Above # Dots per Inch To # Dots per Inch Downsampling is applied to grayscale images with resolution exceeding the specified dots per inch (dpi), reducing the number of dpi to the specified resolution, using the selected Image Sampling method.

NOTE: If the No Downsampling option is selected, the Resolution and Threshold settings do not result in any change in the output file.

-
- ▶ **Image Compression:** There are two types of image compression: lossless and lossy. Lossless compression reduces the file size of an image file, without loss of data, by removing only repetitive information. Lossy compression also reduces file size, but by removing data from the image. When an image is compressed using lossy compression, its visual quality can differ from the original, particularly at higher magnification levels. Because lossy compression discards data, it can achieve smaller file sizes than lossless compression. Compression options include:

No Compression No compression is applied.

JPEG Compression The JPEG (Joint Photographic Experts Group) filter uses a lossy compression technique, encoding images as JPEGs to achieve a smaller file size. This method is typically recommended for continuous-tone photographic images which have more detail than can be reproduced in the desired output device (e.g. monitor or printer).

Flate Compression The Flate filter uses a lossless compression technique and typically produces good compression ratios. Flate is most useful in compressing images containing large areas of single colors or patterns, such as screen shots or simple paint images. Flate is also known as ZIP compression.

Monochrome Image Settings

The Monochrome Image Settings pane includes compression and resampling options for monochromatic images, enabling you to optimize your PDF to achieve the best balance between visual quality and resultant file size

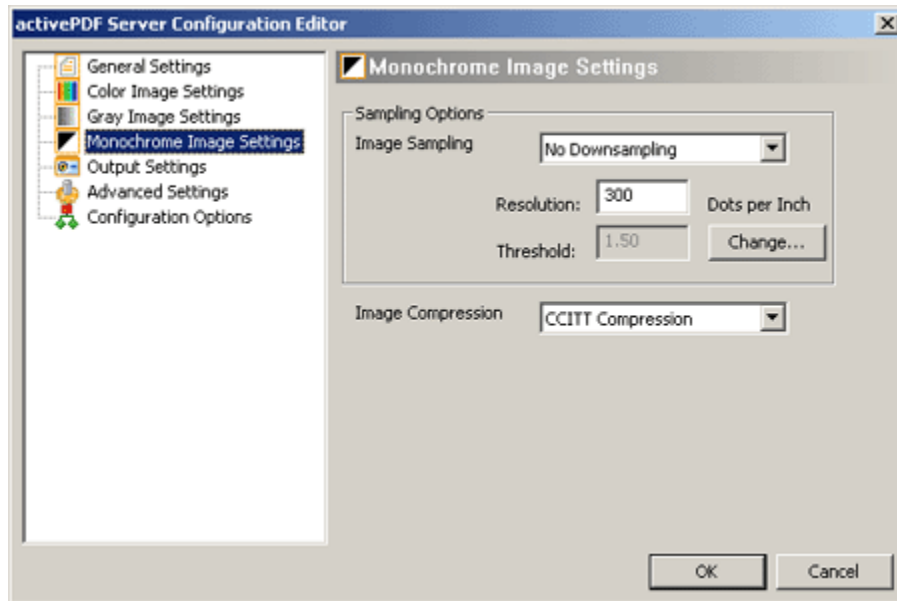


Figure 3.5: Monochrome Image Settings

- Sampling Options:** Sampling options specify downsampling or subsampling for monochromatic images, using the specified resolution parameters. By altering the settings, you can minimize the loss of detail while maximizing reduction in file size. For example, if you are generating a PDF that is intended for onscreen display, you may wish to downsample all images to 72 DPI to reduce file size as well as the time it takes to display the image.

Image Sampling

Downsampling or subsampling reduces the number of dots per inch (DPI) in an image by combining dots in a sample area to create one larger dot. Since there is less information that needs to be saved, the resultant PDF file sizes can be considerably smaller. Sampling options include:

- No Downsampling** No monochromatic images are downsampled in the output PDF file. This option produces the highest quality of output with no impact to file size, and might be used if you are creating PDF files for printing to a commercial-quality printer.
- Average Downsampling** Averages neighboring dots, replacing dots of similar value with the average dot color. Averaging generally produces good quality results and is suitable for printing to most desktop printers, when set to the proper resolution.
- Bicubic** Downsampling occurs using bicubic interpolation to generate new dot values at the specified resolution. In plain language, bicubic interpolation averages the dots in a sample group and replaces the entire group with an average of the dot color at the specified resolution. Bicubic downsampling is more precise, resulting in smoother gradations than other subsampling methods, although the PDF file is slower to generate.
- Subsample** Subsampling takes a dot from the center of a sample group and replaces the entire group with the selected dot at the specified resolution. Subsampling results in

smaller file sizes and faster PDF creation, but you may notice a decrease in visual quality.

- Resolution** The target resolution (measured in dpi) for downsampling monochromatic images using the selected Image Sampling method.
- Threshold** The threshold ratio for downsampling images to the resolution specified. For example, if resolution is set to 300 and threshold is set to 1.5, images in your input file with a resolution above 450 dpi will be downsampled to 300 dpi ($300 * 1.5 = 450$). Click **Change** to calculate the downsampling threshold.

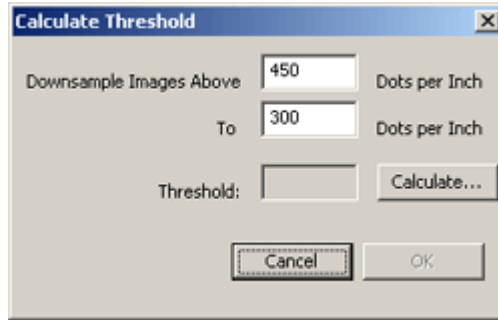


Figure 3.3:Calculate Threshold

Downsample Images Above # Dots per Inch To # Dots per Inch Downsampling is applied to monochromatic images with resolution exceeding the specified dots per inch (dpi), reducing the number of dpi to the specified resolution, using the selected Image Sampling method.
NOTE: If the No Downsampling option is selected, the Resolution and Threshold settings do not result in any change in the output file.

- ▶ **Image Compression:** There are two types of image compression: lossless and lossy. Lossless compression reduces the file size of an image file, without loss of data, by removing only repetitive information. Lossy compression also reduces file size, but by removing data from the image. When an image is compressed using lossy compression, its visual quality can differ from the original, particularly at higher magnification levels. Because lossy compression discards data, it can achieve smaller file sizes than lossless compression. Compression options include:

- No Compression** No compression is applied.
- CCITT Compression** CCITT (International Coordinating Committee for Telephony and Telegraphy) is a lossless compression method which uses standard fax compression to divide the original image into scan lines and then compresses each line based on the previous line. This method generally yields the best results for typical monochromatic images.
- Flate Compression** The Flate filter uses a lossless compression technique and typically produces good compression ratios. <> Flate is also known as ZIP compression.
- RunLength Compression** RunLength is a lossless compression technique which stores consecutive data segments containing identical information as a single data lue and count, to reduce redundant information. For example, an image with large areas of a single color (such as images containing plain black or white backgrounds) will have long `runs` of 0s or 255s. Rather than storing hundreds of 0s, these can be combined into a single element, effectively saying `the following series of 200 have a value of 0`.

Output Settings

The Output Settings pane enables you to specify additional PDF output settings such as automatic rotation for PDF pages, text encoding settings, and linearization.

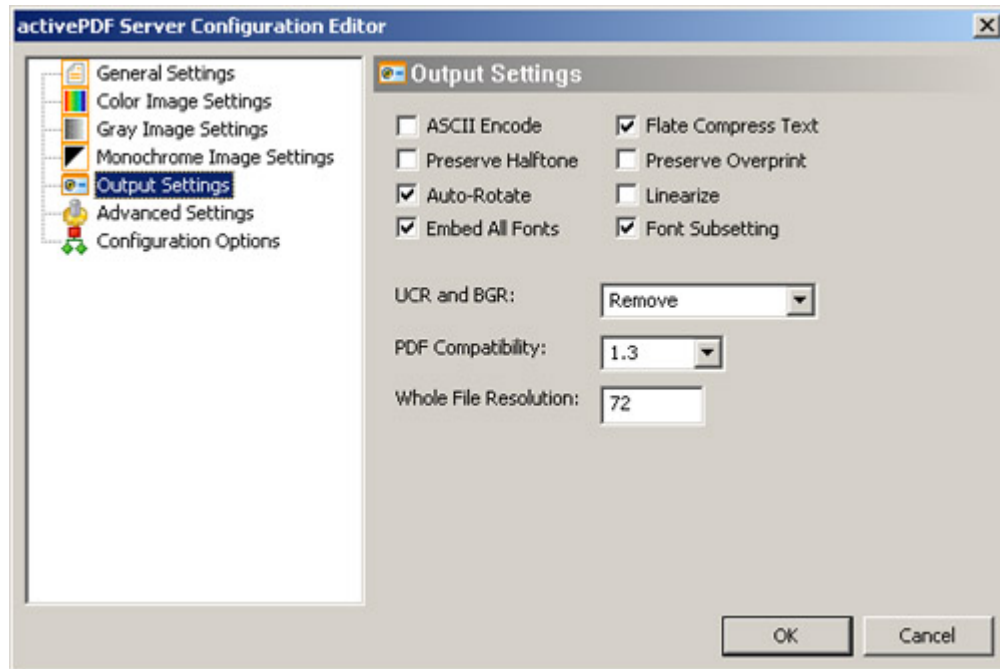


Figure 3.6: Output Settings

- ▶ **ASCII Encode:** If set, activePDF Server ASCII85 encodes binary streams including page content, sampled images and embedded fonts to produce an ASCII PDF. Otherwise the output may contain mostly binary data.

NOTE: This setting cannot be used with Flate Compress Text.

- ▶ **Flate Compress Text:** activePDF Server Flate encodes binary streams or ASCII text producing a mostly binary PDF, which can decrease the overall output size.

NOTE: This setting cannot be used with ASCII Encode.

- ▶ **Preserve Halftone:** Retains any existing halftone information in files being converted to PDF. Halftone images are comprised of series of dots in a specific pattern that simulates the look of a continuous tone image. Since printers cannot print continuous tones, the halftone information controls the amount of ink that is deposited on the paper for each color, to achieve the best blend in color deposit. Halftone information is typically intended for use with a particular printer.
- ▶ **Preserve Overprint:** Retains any existing overprint settings in files being converted to PDF. Overprint occurs when two or more inks print on top of each other to produce a combined color or desired color effects; Failing to preserve the overprint results in only one color being printed. Overprint information is intended for use with specific output devices.
- ▶ **Auto-Rotate:** Automatically rotates pages based on the orientation of the majority of the text.

- ▶ **Linearize:** Linearization, or web optimization, prepares the output for byte-serving, enabling page-at-a-time downloading when served through a web browser.

NOTE: Text and line art compression occur regardless of compression settings.

- ▶ **Embed All Fonts:**

Embeds all fonts used in the input file. Font embedding enables you to include the fonts used in your document in the resultant PDF; this means that users do not need to have these fonts installed in order to properly display the PDF.

NOTE: You should check the terms of your font licensing agreement(s) before embedding any fonts to verify that this use is permitted.

- ▶ **Font Subsetting:** When set, if the percentage of characters from a given font is less than 60%, activePDF Server subsets the font(s) to include only those characters used in the input file. Subsetting results in a smaller file size than the “Embed All Fonts” option.

NOTE: Font subsetting may limit a user’s ability to edit text in the PDF document, since some characters may be missing.

- ▶ **UCR and BGR:** In printing, Undercolor Removal (UCR) reduces the amount of Cyan, Magenta, and Yellow ink to compensate for the Black Generation (BGR). BGR is the amount of black ink necessary to reproduce a particular color. UCR and BGR information is used with specific output devices. Options include:

Preserve	Retains any existing UCR or BGR settings in files being converted to PDF. <i>(Default)</i>
Remove	Removes any existing UCR or BGR settings.

- ▶ **PDF Compatibility:** Creates PDF version 1.2, 1.3, 1.4 or 1.5 files, for compatibility with PDF viewing and editing applications. Options include:

- ▶ PDF version 1.2 (Adobe® Acrobat 3.x)
- ▶ PDF version 1.3 (Adobe Acrobat 4.x)
- ▶ PDF version 1.4 (Adobe Acrobat 5.x)
- ▶ PDF version 1.5 (Adobe Acrobat 6.x)

NOTE: Depending on the option you choose, certain features may not be supported in your outputted PDF file. For example, 128-bit encryption is supported in PDF 1.4 and later.

- ▶ **Whole File Resolution:** The resolution to be used in producing the output PDF, expressed in Dots Per Inch (DPI). Whole file resolution controls the resolution for pattern fills and for fonts that must be converted to bitmaps, and does not affect image resolution settings. In general, Whole File Resolution should be set to match the resolution of the intended output device. Default is 72, which is suitable for onscreen display.

Advanced Settings

The Advanced Settings specify additional process controls and output options. You can experiment with the settings to find the best level of performance for your implementation.

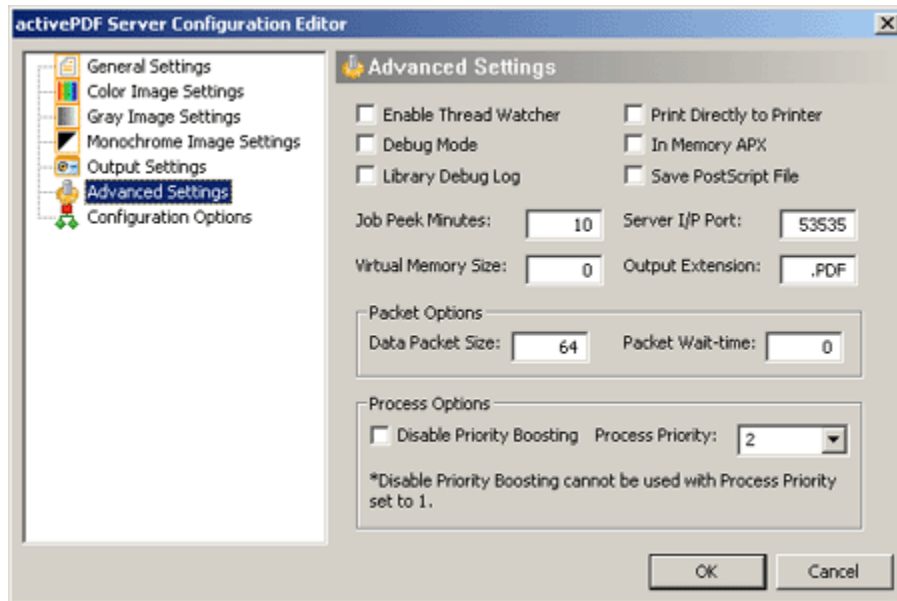


Figure 3.8: Advanced Settings

- ▶ **Enable Thread Watcher:** activePDF Server generates an additional watcher thread that provides increased monitoring of the process to help prevent threads that fail from hanging indefinitely.
- ▶ **Print Directly to Printer:** activePDF Server prints the document directly to PDF and does not spool. This may help in processing large documents, as it eliminates the need for the temporary copy of the file that is created as part of the spooling process.
- ▶ **Debug Mode:** activePDF Server writes process details to the APSPDF.log file in the Windows or WinNT System32 directory.

NOTE: This option should be left unchecked unless advised by activePDF Technical Support. After gathering the necessary log files, be sure to uncheck this option to resume normal document processing.
- ▶ **In Memory APX:** Typically, activePDF Server generates a Document Configuration Profile (.APX) file for each job in the Windows or WinNT APSDoc directory. When this option is selected, the APX file generates IN-MEMORY, instead of being written to disk.
- ▶ **Library Debug Log:** Generates a log file containing information regarding the file being converted, along with the conversion parameters. This file is saved in the directory specified by the Default Directory setting.

NOTE: This option should be left unchecked unless advised by activePDF Technical Support. After gathering the necessary log file, be sure to uncheck this option to resume normal document processing.
- ▶ **Save PostScript File:** Saves the intermediate PostScript file (used in generating the output PDF) in the default queue

directory (C:\WINDOWS\activePDF).

- ▶ **Job Peek Minutes:** The maximum time (in seconds) the activePDF Server service sleeps before checking the printer currently being used in the Print to PDF operation. If the printer is active, the service sleeps again; otherwise, it returns the printer to the pool. Default is 10.
- ▶ **Server I/P Port:** The port required for the activePDF Server print to PDF operation. Default is 53535.

NOTE: This port must remain open and dedicated to activePDF Server.

- ▶ **Virtual Memory Size:** The amount of virtual memory (in bytes) allocated to the activePDF Server process during a print to PDF operation. Increasing this number may help in processing large documents.
- ▶ **Output Extension:** The file extension applied to output generated by activePDF Server. Default is .PDF.

NOTE: This does not set the file type. The dot (.) before file extension is required.

- ▶ **Packet Options:** The Packet Options specify the data packet size and packet sleep time.

NOTE: Packet Options settings are used in the activePDF Server communications layer. Adjusting these settings is not recommended. Packet options include:

Data Packet Size	Data packet size is the size (in kilobytes) of the buffer used by activePDF Server during communication. Defaults is 64.
Packet Wait-time seconds	The time the activePDF Server process sleeps between sending each data packet. Options include: 0 = No Wait. (<i>Default</i>) -1 = Yields all time to waiting processes before continuing.

- ▶ **Process Options:** The Process Options specify the process priority and boosting priority. Process options include:

Disable Priority Boosting	Overrides the default operating system functionality to increase process priority level based on load. CAUTION: This option cannot be used if Process Priority is set to "1". This will prevent activePDF Server from processing print jobs.
Process Priority	The level of process importance used by the operating system to determine processing order. Options include: 0 Normal Level: Processed in the order received after higher level processes are = complete. 1 Idle Level: Only processed during idle periods. = 2 High Level: Always processed before lower priority processes. =

Configuration Options

The Configuration Options enable you to save your current configuration, import a previous configuration, or restore the activePDF Server Configuration Editor to the default settings.

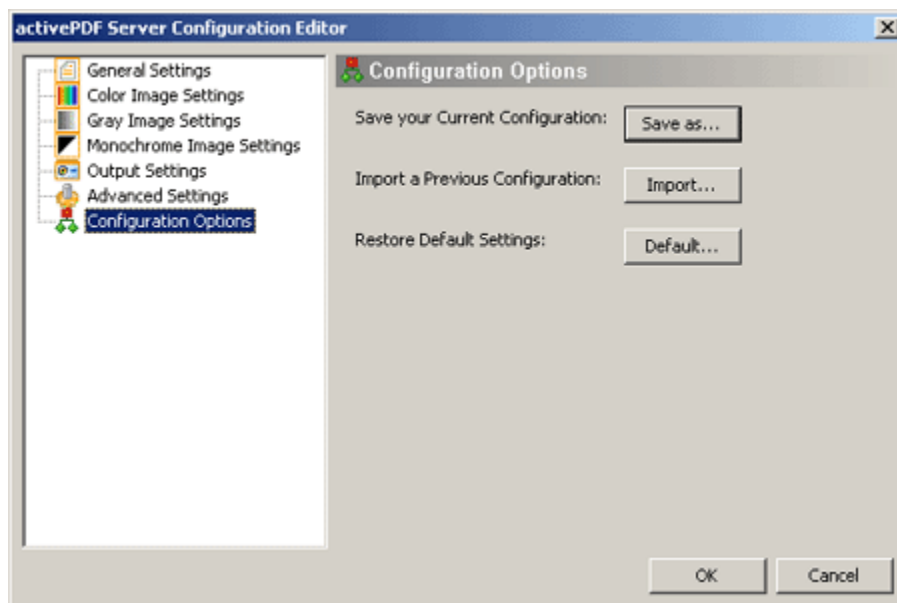


Figure 3.9: Configuration Options

- ▶ **Save your Current Configuration:** Exports the current activePDF Server Configuration Editor settings as a file (.APS). This file is used for importing or with [LoadConfiguration](#).
- ▶ **Import a Previous Configuration:** Import a previously saved activePDF Server Configuration Editor settings (.APS) file.

NOTE: Does not reset the [Terminal Server Settings](#).

- ▶ **Restore Default Settings:** Returns the activePDF Server Configuration Editor to the default settings.

NOTE: Does not reset the [Terminal Server Settings](#).

Terminal Server Settings

The Terminal Server Settings specify the location of the Document Configuration file (.APX) and root Windows or WinNT directory for Microsoft® Terminal Services installed in Application mode. In Application mode, the Current User session prevents activePDF Server from accessing required default paths.

NOTE: This screen appears only when Terminal Services is detected in Application mode.

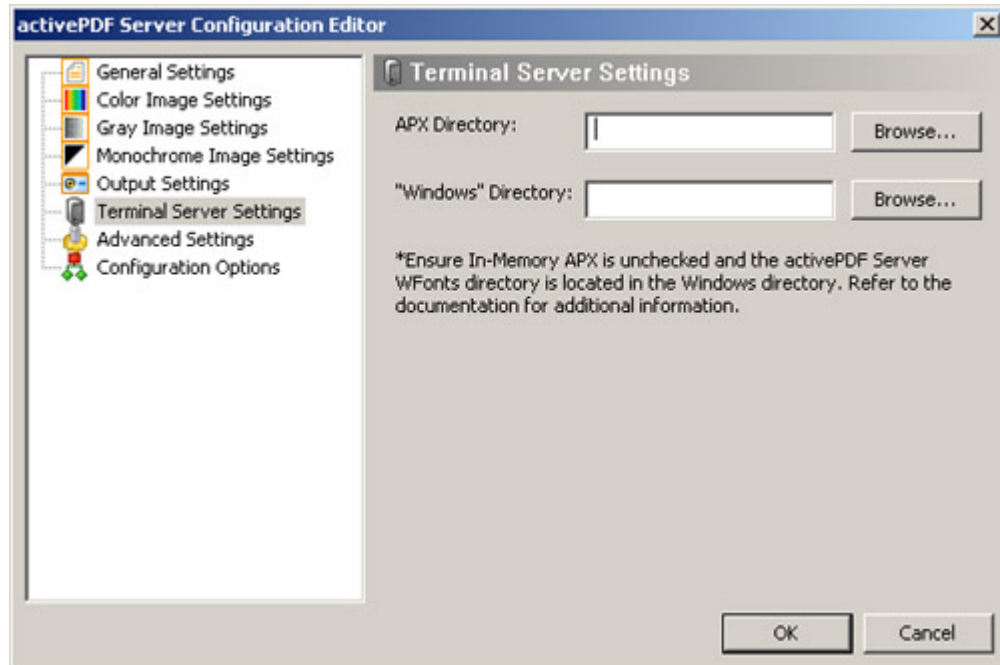


Figure 3.7: Terminal Server Settings

- ▶ **APX Directory:** The location in which the activePDF Server APX file is to be stored.

NOTE: This location must be accessible for the Current User session.

NOTE: In Memory APX must remain unchecked.

- ▶ **"Windows" Directory:** The path to the current User's Windows or WinNT directory.

NOTE: This is not the root Windows or WinNT directory.

Additional Requirements

activePDF Server generates the WFonts directory in the root Windows or WinNT directory. The WFonts directory is activePDF Server's font repository which is required for converting files to PDF. In Application Mode, you must copy the WFonts directory from the root Windows or WinNT directory to the Current User Windows or WinNT directory.

NOTE: You must restart the activePDF Server service after copying or otherwise changing the WFonts directory.

Default Printer

activePDF Server requires a *non-networked* PostScript® printer assigned as the **Default Printer to Emulate** to generate a PDF. By default, Server installs and assigns *activePDF PostScript Printer* as this printer. If something prevents the *activePDF PostScript Printer* from installing, Server tries to detect a local PostScript printer to assign as the **Default Printer to Emulate**. If these actions are prevented during installation, you will need to install and assign the *activePDF PostScript Printer* manually.

1. Log on as the **Local Administrator**.
 2. Download the following files:
 - ▶ <http://www.activepdf.com/AltDownloads/Updates/aps5eng.exe>
 - ▶ <http://www.activepdf.com/AltDownloads/Updates/activepdf.ppd>
 3. Click **Start > Run**, type **X:\WINDOWS\system32\spool\drivers\w32x86** or **X:\WINNT\system32\spool\drivers\w32x86** depending on your installed OS, and then click **OK**.
 4. Move the **activePDF.ppd** file to this location.
 5. Double-click **aps5eng.exe** to run the installation.
 6. When prompted, select the **activePDF.ppd**.
 7. Click **Start > Run**, type **control printers**, and then click **OK**.
 8. Right-click the **activePDF PostScript Printer**, select **Set as Default Printer**.
 9. Restart your machine.
-

Printer & Pool Types

activePDF Server can install two to three PostScript® printers depending on your operating system. The following printers are installed:

- ▶ **activePDF PostScript Printer:** The default activePDF Server printer, which uses the *activePDF Postscript5 Driver*.
- ▶ **activePDF PostScript (4) Printer:** The secondary default activePDF Server printer, which uses the *activePDF Postscript4 Driver*.
- ▶ **activePDF Shadow Printer:** The standard printer for PostScript to PDF operations. Uses the *activePDF Postscript5 Driver*

Server also generates dynamic printers in the dynamic printer pool. Dynamic printers are the most commonly used printers and are generated incrementally as needed during conversion up to the maximum number specified. The printers are maintained internally by activePDF Server and can remain in the *Printers and Faxes* for future use. **Standard Pool Size** controls the size of the dynamic printer pool. An example of a dynamic printer name is *activePDF-5*. Dynamic printers are deleted upon service restart and do not survive reboots. The number of printers should be set to the number of simultaneous users.

NOTE: Server depends on the *OS Spooler*, which has a built-in limit for the number of ports and users. We strongly recommend testing your application to determine what value works best for your machine. In general, the number of dynamic printers should remain below 80. Do not assign *Dynamic* printers as the **Printer To Emulate**.

System Event Log

During standard operations, Server creates and deletes *Dynamic Printers*. Each time this happens, an event is generated in the *Windows System Event Log*. To prevent the log from becoming too large and crashing the system, you might need to ensure that **Do not overwrite events** is not selected.

1. Log on as the **Local Administrator**.
 2. Click **Start > Run**, type **eventvwr.msc**, and then click **OK**.
 3. Under **Event Viewer (Local)**, right-click **System**, and then select **Properties**.
 4. On the **General** Tab, ensure that the **Do not overwrite events** is **Not** selected.
-

Object Reference

activePDF Server is controlled via a programmable COM object. This section contains the technical documentation for the [Methods](#) and [Properties](#) defined in the Server COM object. Once you have become familiar with [Instantiating the Server Object](#), you can use this as your main resource for understanding Server.

- ▶ [Instantiating the Object](#)
- ▶ [Methods](#)
- ▶ [Properties](#)

Instantiating the Object

The Server COM object is represented by the *Class ID* `APServer.Object`. You can use this *Class ID* to instantiate the object.

Instantiation

The instantiation of the Server COM object using the *Class ID* depends on your scripting environment. We recommend the use of *Late-Binding* as the Server type library is already registered in the *Object Registry*. The following VBScript example demonstrates the instantiation process.

```
' Instantiate Object
Set objSVR = CreateObject("APServer.Object")
```

Clean-Up

Certain scripting environments support clearing the variables. Destroying the object prevents the memory build-up after the conversion is finished. Below is a VBScript example that would appear at the end of your code.

```
' Release Object
Set objSVR = Nothing
```

Methods

The methods found in the Server object control several necessary functions, such as the command to begin printing, to those that enable output encryption. The following is the complete list of Server's methods.

- **AbortPrinting**
- **AddFileBookmark**
- **AddLinkedPDFBookmark**
- **AddPageBookmark**
- **AddPDFMark**
- **AddURLBookmark**
- **ClearPDFMarks**
- **DeleteFile**
- **FromString**
- **GetBinaryImage**
- **GetExclusiveAccess**
- **ImageToPDF**
- **ImpersonateUser**
- **IsFingerprintValid**
- **LoadConfiguration**
- **LoadImageInfo**
- **LoadRemoteProfile**
- **PDFToBrowser**
- **PrintToBrowser**
- **PSToPDF**
- **SetColorDownsampleThreshold**
- **SetGrayDownsampleThreshold**
- **SetMonoDownsampleThreshold**
- **SetOutputSecurity**
- **SetOutputSecurity128**
- **SetPrinterAsDefault**
- **SetViewMode**
- **StartPrinting**
- **StopImpersonation**
- **StopPrinting**
- **Test**
- **ToString**
- **Wait**
- **XMLData**

AbortPrinting

Description: Instructs Server to terminate the in-progress print job and returns the printer to the printer pool.

Syntax: `retValue = object.AbortPrinting`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	<p>A value indicating the outcome of the operation.</p> <p>0 = Success.</p> <p># = An activePDF Server error.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

Remarks: This method is only used when the application prematurely stops printing. Otherwise, the printer remains idle until the *activePDF Server Service* is restarted. [StopPrinting](#) should be used in all other instances. By default, the unique identifier is applied as the output filename unless [NewDocumentName](#) is called.

AddFileBookmark

Description: Adds a bookmark to the output PDF that links to an external document.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.AddFileBookmark "BookmarkName", NbrChildBookmarks, "ExternalFileName"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	BookmarkName	STRING	The name of the bookmark.
	NbrChildBookmarks	LONG	The number of sub-level bookmarks found under the bookmark in the document tree. Defaults to 0 (<i>zero</i>). For example: <pre>APServer.AddFileBookmark "Parent", 2, "C:\Parent.txt" APServer.AddFileBookmark "Child 1", 0, "C:\Child1.txt" APServer.AddFileBookmark "Child 2", 0, "C:\Child2.txt"</pre>
	ExternalFileName	STRING	The full path and file name of the external document.

AddLinkedPDFBookmark

Description: Adds a bookmark to the output PDF that links to an external PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.AddLinkedPDFBookmark "BookmarkName", NbrChildBookmarks, "PDFFileName", PageNbr, "PDFView"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	BookmarkName	STRING	The name of the bookmark.
	NbrChildBookmarks	LONG	The number of sub-level bookmarks found under the bookmark in the document tree. Defaults to 0 (zero). For example: <pre>APServer.AddLinkedPDFBookmark "Parent", 2, "C:\doc1.pdf", 2, "Fit" APServer.AddLinkedPDFBookmark "Child 1", 0, "C:\doc1.pdf", 2, "Fit" APServer.AddLinkedPDFBookmark "Child 2", 0, "C:\doc1.pdf", 2, "Fit"</pre>
	PDFFileName	STRING	The filename and path of the destination PDF file.
	PageNbr	LONG	The page in the destination PDF to display.
	PDFView	STRING	The PDF Views used to display the page in the destination PDF.

AddPageBookmark

Description: Adds a bookmark to the output PDF that links to a page in the PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.AddPageBookmark "BookmarkName", NbrChildBookmarks, PageNbr, "PDFView"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	BookmarkName	STRING	The name of the bookmark.
	NbrChildBookmarks	LONG	The number of sub-level bookmarks found under the bookmark in the document tree. Defaults to 0 (<i>zero</i>). For example: <pre>APServer.AddPageBookmark "Parent", 2, 2, "Fit" APServer.AddPageBookmark "Child 1", 2, 0, "Fit" APServer.AddPageBookmark "Child 2", 2, 0, "Fit"</pre>
	PageNbr	LONG	The PDF page opened when clicking the bookmark.
	PDFView	STRING	One of the available PDF Views to be placed in the output PDF.

AddPDFMark

Description: Adds a **PDFMark** to be placed in the output PDF during conversion.

NOTE: Must be called prior to **StartPrinting**. Requires a strong understanding of PostScript® code.

Syntax: `object.AddPDFMark "PDFMark"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	PDFMark	STRING	One of the available PDFMarks to place in the output PDF.

AddURLBookmark

Description: Adds a bookmark to the output PDF that links to an external URL.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.AddURLBookmark "BookmarkName", NbrChildBookmarks, "DestinationURL"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	BookmarkName	STRING	The name of the bookmark.
	NbrChildBookmarks	LONG	The number of sub-level bookmarks found under the bookmark in the document tree. Defaults to 0 (<i>zero</i>). For example: <div data-bbox="708 894 1484 1041" style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> <pre>APServer.AddURLBookmark "Parent", 2, "www.activepdf.com" APServer.AddURLBookmark "Child 1", 0, "www.activepdf.com" APServer.AddURLBookmark "Child 2", 0, "www.activepdf.com"</pre> </div>
	DestinationURL	STRING	The URL opened upon clicking the bookmark.

ClearPDFMarks

Description: Clears all **PDF Marks** before they are applied to the output PDF.

NOTE: Must be called prior to **StartPrinting**.

Syntax: `object.ClearPDFMarks`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

DeleteFile

Description: Deletes the specified PDF.

NOTE: Must be called after [StopPrinting](#)

Syntax: `retValue = object.DeleteFile ("PDFFileName")`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	PDFFileName	STRING	The name and path of the PDF to delete.

retValue:	Value Type	Description & Settings
	LONG	A value indicating the outcome of the operation. 0 = Success. # = A Win32 error code.

FromString

Description: Converts the activePDF WebGrabber™ [Prt2DiskSettings](#) into parameters retrievable by the [Server properties](#).

NOTE: Must call after the WebGrabber [Wait](#) method.

Syntax: `object.FromString = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The WebGrabber Prt2DiskSettings .

GetBinaryImage

Description: Returns a binary image of the input PDF, which is suitable for display to a browser.

Syntax:

```
retValue = object.GetBinaryImage ("InputFilename" )
```

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	InputFilename	STRING	The full path and filename for the input PDF of which to return a binary image.

retValue:	Value Type	Description & Settings
	BINARY	The input PDF as binary data.

GetExclusiveAccess

Description: Assigns exclusive access to the PostScript® file for a **PSToPDF** operation.

Syntax: `retValue = object.GetExclusiveAccess ("PSFileName", TimeoutSeconds)`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	PSFileName	STRING	The name and path of the PS to convert.
	TimeoutSeconds	INTEGER	The maximum wait time in seconds before retrieving status. Defaults to 30.

retValue:	Value Type	Description & Settings
	INTEGER	A value indicating the outcome of the operation. 0 = Success. # = An activePDF Server error.
		NOTE: Contact Technical Support if 0 is not returned

ImageToPDF

Description: Instructs Server to convert the specified image file to PDF.

NOTE: Refer to the [ImageInfoFileType](#) property for a list of image types.

NOTE: Certain parameters of this method have been deprecated in activePDF Server 3.8 and will no longer perform the described function. Refer to the below table for a list of the deprecated parameters.

Syntax: `retValue = object.ImageToPDF ("ImageFileName", "PDFFileName", AutoRotate, Resolution)`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
ImageFileName	STRING	The name and path for the image to convert.
PDFFileName	STRING	The name and path for the output PDF.
Width	INTEGER	The width of the image in PDF Units . Deprecated 0 = Retains image width. <i>(Default)</i>
Height	INTEGER	The height of the image in PDF Units . Deprecated 0 = Retains image height. <i>(Default)</i>
PersistRatio	BOOLEAN	Indicates how the image should be adjusted to the width and height. Deprecated True = Image adjusted proportionally to width and height. False = Image stretches to width and height. <i>(Default)</i>
AutoRotate	BOOLEAN	Indicates if image should be rotated based on text direction. True = Image will be rotated based on text direction. False = Image retains original rotation. <i>(Default)</i>
Resolution	INTEGER	The resolution of the image in Dots Per Inch (<i>DPI</i>). Default is 72.
PaperSize	STRING	A value indicating the size of the output PDF. Deprecated <i>A0, A1, A2, A3, A4, A5, A6, B5, LETTER, LEDGER, LEGAL, T11X17</i>
OffsetX	INTEGER	The horizontal offset origin from 0, 0 for the lower-left corner of the image in the output PDF using <i>PDF Coordinates</i> . Deprecated
OffsetY	INTEGER	The vertical offset origin from 0, 0 for the lower-left corner of the image in

the output PDF using *PDF Coordinates*. **Deprecated**

retValue:	Value Type	Description & Settings
LONG		A value indicating the outcome of the operation. +# = A Windows® error. 0 = Success. -# = An activePDF Server error. NOTE: Contact Technical Support if 0 is not returned

ImpersonateUser

Description: Designates a user account for Server to impersonate when an application is running under the *SYSTEM* account and cannot access the [Default Printer to emulate](#).

Syntax: `retValue = object.ImpersonateUser ("Domain", "UserID", "Password")`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
Domain	STRING	The domain. Defaults to "." for the local account database. Pass "" to use the value specified in the DefaultDomain string value or DefaultDomain property.
UserID	STRING	The case-sensitive user ID for the account. Defaults to "". Pass "" to use the value specified in the DefaultUser string value or DefaultUser property.
Password	STRING	The case-sensitive password for the account. Defaults to "". Pass "" to use the value specified in the DefaultPassword string value or DefaultPassword property.

Value Type	Description & Settings
LONG	A value indicating the outcome of the operation. 0 = Success. # = Unsuccessful.

IsFingerprintValid

Description: Validates the integrity of an activePDF Fingerprint added with [FingerprintPDF](#).

Syntax: `retValue = object.IsFingerprintValid ("PDFFileName")`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	PDFFileName	STRING	The name and path of the PDF to check.

retValue:	Value Type	Description & Settings
	BOOLEAN	<p>A value indicating the integrity of the fingerprint.</p> <p>True = Unchanged.</p> <p>False = The fingerprint has been changed.</p>

LoadConfiguration

Description: Instructs Server to use the specified [Document Configuration Profile](#) (APS) to generate a PDF.

Syntax: `retValue = object.LoadConfiguration ("APSFileName")`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	APSFileName	STRING	The name and path of the APS file to use.

retValue:	Value Type	Description & Settings
	LONG	A value indicating the outcome of the operation. 0 = Success. # = A Win32 error code.

LoadImageInfo

Description: Retrieves the file information from the image file specified with [ImageToPDF](#) for use with [ImageInfoBitsPerPixel](#), [ImageInfoFileType](#), [ImageInfoHeight](#), [ImageInfoWidth](#), [ImageInfoXResolution](#), [ImageInfoYResolution](#), and [ImageInfoTotalPages](#).

Syntax: `retValue = object.LoadImageInfo ("ImageFileName")`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	ImageFileName	STRING	The name and path of the image.

retValue:	Value Type	Description & Settings
	LONG	<p>A value indicating the outcome of the operation.</p> <p>1 = Success.</p> <p><>1 = An error has occurred.</p> <p>NOTE: Contact Technical Support if 1 is not returned</p>

LoadRemoteProfile

Description: During remote submission, passes the remote IP Address to the machine where Server is installed loading a [Document Configuration Profile](#).

NOTE: Only used with remote submissions.

Syntax: `retValue = object.LoadRemoteProfile (IPAddress)`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	IPAddress	LONG	The IP address.

retValue:	Value Type	Description & Settings
	LONG	<p>A value indicating the outcome of the operation.</p> <p>0 = Success.</p> <p><>0 = An error has occurred.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

PDFToBrowser

Deprecated: Use [GetBinaryImage](#) with [DeleteFile](#) instead.

Description: Displays the output PDF in a browser window. Might require a response buffer.

Syntax: `object.PDFToBrowser "OutputPath", DeleteWhenDone`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	OutputPath	STRING	The path to the output PDF.
	DeleteWhenDone	SHORT	Instructs Server to delete the output PDF when the browser is closed. 0 = File is not deleted. (<i>Default</i>) 1 = File is deleted.

PrintToBrowser

Deprecated: Use [GetBinaryImage](#) with [DeleteFile](#) instead.

Description: Displays the output PDF to a browser window.

Syntax: `object.PrintToBrowser "OutputPath" , "ContentType" , DeleteWhenDone`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	OutputPath	STRING	The path to the output PDF.
	ContentType	STRING	The content type for the PDF per HTML standards.
	DeleteWhenDone	SHORT	Instructs Server to delete the output PDF when the browser is closed. 0 = File is not deleted. <i>(Default)</i> 1 = File is deleted.

PSToPDF

Description: Instructs Server to convert the specified PostScript® file to PDF.

Syntax: `retValue = object.PSToPDF ("PSFileName" , "PDFFileName")`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	PSFileName	STRING	The name and path of the PS to convert.
	PDFFileName	STRING	The name and path of the output PDF

retValue:	Value Type	Description & Settings
	LONG	<p>A value indicating the outcome of the operation.</p> <p>0 = Success.</p> <p><>0 = An error has occurred.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

SetColorDownsampleThreshold

Description: The threshold ratio for downsampling images to the resolution specified. For example, if resolution is set to 300 and threshold is set to 1.5, images in the input file with a resolution above 450 dpi will be downsampled to 300 dpi ($300 * 1.5 = 450$).

NOTE: `ColorDownsampleType` must be set.

Syntax: `object.SetColorDownsampleThreshold (originalPPI,outputResolution)`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	originalPPI	LONG	Downsample all images meeting or exceeding this value to the outputResolution. This value must be at least as great as the outputResolution value, but no greater than ten times the outputResolution value. For example, if the outputResolution is 72, the available value ranges from 72 to 720.
	outputResolution	LONG	The target output resolution.

SetGrayDownsampleThreshold

Description: The threshold ratio for downsampling images to the resolution specified. For example, if resolution is set to 300 and threshold is set to 1.5, images in the input file with a resolution above 450 dpi will be downsampled to 300 dpi ($300 * 1.5 = 450$).

NOTE: `GrayDownsampleType` must be set.

Syntax: `object.SetGrayDownsampleThreshold (originalPPI,outputResolution)`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	originalPPI	LONG	Downsample all images meeting or exceeding this value to the outputResolution. This value must be at least as great as the outputResolution value, but no greater than ten times the outputResolution value. For example, if the outputResolution is 72, the available value ranges from 72 to 720.
	outputResolution	LONG	The target output resolution.

SetMonoDownsampleThreshold

Description: The threshold ratio for downsampling images to the resolution specified. For example, if resolution is set to 300 and threshold is set to 1.5, images in the input file with a resolution above 450 dpi will be downsampled to 300 dpi ($300 * 1.5 = 450$).

NOTE: `MonoDownsampleType` must be set.

Syntax: `object.SetMonoDownsampleThreshold (originalPPI,outputResolution)`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	originalPPI	LONG	Downsample all images meeting or exceeding this value to the outputResolution. This value must be at least as great as the outputResolution value, but no greater than ten times the outputResolution value. For example, if the outputResolution is 72, the available value ranges from 72 to 720.
	outputResolution	LONG	The target output resolution.

SetOutputSecurity

Description: Adds 40-bit encryption to the output PDF.

NOTE: Must be called prior to [StartPrinting](#). Options might not be supported by all PDF viewers.

Syntax: `object.SetOutputSecurity "UserPassword", "OwnerPassword", CanPrint, CanEdit, CanCopy, CanModify`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
UserPassword	STRING	Case-sensitive password required to view document, which has a 32-character maximum and cannot contain control characters. NOTE: The password cannot be changed once set. For evaluations, the prefix DEMO will be added and counts towards the character limit. For example, the password TEST becomes DEMOTEST.
OwnerPassword	STRING	Case-sensitive password required to modify document, which has a 32-character maximum and cannot contain control characters. NOTE: The password cannot be changed once set. For evaluations, the prefix DEMO will be added and counts towards the character limit. For example, the password TEST becomes DEMOTEST.
CanPrint	LONG	Enables or disables printing for the output PDF. True = Enable printing. <i>(Default)</i> False = Disable printing.
CanEdit	BOOLEAN	Enables or Disables editing for the output PDF. True = Enable editing <i>(Default)</i> False = Disable editing.
CanCopy	BOOLEAN	Enables or disables copying of text and graphics for the output PDF. True = Enable copying of text and graphics. <i>(Default)</i> False = Disable copying of text and graphics.
CanModify	BOOLEAN	Enable or disable document modification in the output PDF. True = Enable modification. <i>(Default)</i> False = Disable modification.

SetOutputSecurity128

Description: Adds 128-bit encryption to the output PDF.

NOTE: Must be called prior to [StartPrinting](#). Options might not be supported by all PDF viewers.

Syntax: `object.SetOutputSecurity "UserPassword", "OwnerPassword", CanPrint, CanEdit, CanCopy, CanModify, CanFillInFormFields, CanMakeAccessible, CanAssemble, CanReproduce`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	UserPassword	STRING	Case-sensitive password required to view document, which has a 32-character maximum and cannot contain control characters. NOTE: The password cannot be changed once set. For evaluations, the prefix DEMO will be added and counts towards the character limit. For example, the password TEST becomes DEMOTEST.
	OwnerPassword	STRING	Case-sensitive password required to modify document, which has a 32-character maximum and cannot contain control characters. NOTE: The password cannot be changed once set. For evaluations, the prefix DEMO will be added and counts towards the character limit. For example, the password TEST becomes DEMOTEST.
	CanPrint	LONG	Enables or disables printing for the output PDF. True = Enable printing. <i>(Default)</i> False = Disable printing.
	CanEdit	BOOLEAN	Enables or Disables editing for the output PDF. True = Enable editing <i>(Default)</i> False = Disable editing.
	CanCopy	BOOLEAN	Enables or disables copying of text and graphics for the output PDF. True = Enable copying of text and graphics. <i>(Default)</i> False = Disable copying of text and graphics.
	CanModify	BOOLEAN	Enable or disable document modification in the output PDF. True = Enable modification. <i>(Default)</i> False = Disable modification.

CanFillInFormFields	BOOLEAN	Enable or disable form field population in the output PDF. True = Enable form field population. <i>(Default)</i> False = Disable form field population.
CanMakeAccessible	BOOLEAN	Enables or disables the use of any accessibility aids added to the output PDF with an external application. True = Enable the use of accessibility aids. <i>(Default)</i> False = Disable the use of accessibility aids.
CanAssemble	BOOLEAN	Enables or disables document assembly for output PDF. True = Enable user to insert, rotate, delete pages, generate bookmarks and thumbnails. <i>(Default)</i> False = Disable document assembly. NOTE: Supersedes <code>CanModify</code>
CanReproduce	BOOLEAN	Enables or disables reproduction of output PDF. True = Enable faithful reproduction. <i>(Default)</i> False = Disable reproduction. NOTE: If <i>False</i> and <code>CanPrint</code> is set to <i>True</i> , low-resolution printing is enabled.

SetPrinterAsDefault

Description: Sets the printer specified in [NewPrinterName](#) as the current default printer. Used with applications that must use the default printer.

Syntax: `object.SetPrinterAsDefault`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

SetViewMode

Description: Sets the initial view of your PDF.

NOTE: Must be called after [StopPrinting](#).

Syntax: `object.SetViewMode InitialViewMode, PageNbr, "PDFView"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	InitialViewMode	LONG	The initial view mode for the output PDF. 0 = Normal Uses the default settings. 1 = Show Outlines Bookmarks panel is visible and all child bookmarks are expanded. 2 = Show Thumbnails Pages panel is visible. 3 = Full Screen PDF opens in full screen mode.
	PageNbr	LONG	The page displayed on initial view.
	PDFView	STRING	One of the available PDF Views to apply to the output PDF.

StartPrinting

Description: Starts the PDF generation process by initializing the Server Service.

NOTE: Must be called after all necessary or desired properties.

Syntax: `retValue = object.StartPrinting`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	LONG	<p>A value indicating the outcome of the operation.</p> <p>+ # = A Windows® error.</p> <p>0 = Success.</p> <p>- # = An activePDF Server error.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

StopImpersonation

Description: Reverts the current user to the state prior to calling [ImpersonateUser](#).

NOTE: This might revert your security context to a different level.

Syntax: `object.StopImpersonation`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

StopPrinting

Description: Ensures proper completion of print cycle. Call after automating an application to print.

Syntax: `object.StopPrinting`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

Test

Description: Instructs Server to generate a *Test* output in the *Default Output Path*. Useful for verifying that the software is installed and running correctly.

Syntax: `retValue = object.Test ("PDFFileName")`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	PDFFileName	STRING	The name and path of the PDF to check.

retValue:	Value Type	Description & Settings
	LONG	<p>A value indicating the outcome of the operation.</p> <p>+ # = A Windows® error.</p> <p>0 = Success.</p> <p>- # = An activePDF Server error.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

ToString

Description: Returns the job configuration settings as a string. Required to use Server with DocConverter™, WebGrabber™, Printer™.

NOTE: Must be called prior to [StartPrinting](#). In .NET, use the [XMLData](#) method instead.

Syntax: `"retValue" = object.ToString`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	STRING	The job configuration settings as an XML string.

Wait

Description: Returns the status of Server's operations by polling using the internal conversation ID.

NOTE: This is done using alternate *sleeps* to prevent overloading of the CPU.

Syntax: `retValue = object.Wait (WaitSeconds)`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	WaitSeconds	INTEGER	The maximum wait time for retrieving status in seconds (<i>By default, 30</i>).

retValue:	Value Type	Description & Settings
	LONG	A value indicating the outcome of the operation. -99 = Operation Timeout. 0 = Success.

XMLData

Description: Returns the job configuration settings as a string in .NET. Required to use Server with DocConverter™, WebGrabber™, Printer™.

NOTE: Must be called prior to [StartPrinting](#). If you are not using .NET, use the [ToString](#) method instead.

Syntax: `"retValue" = object.ToString`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	STRING	The job configuration settings as an XML string.

Properties

The Server object's properties are used to set several key output parameters, including resolution, page size and document information.

- **ASCIIEncode**
- **AutoRotate**
- **ChunkSize**
- **ColorDownsampleType**
- **ColorFilter**
- **ColorImageFlags**
- **ColorImageResolution**
- **CompatabilityLevel**
- **ConcatenateToExisting**
- **DefaultDomain**
- **DefaultPassword**
- **DefaultPrinter**
- **DefaultQueue**
- **DefaultUser**
- **DocumentName**
- **EmbedAllFonts**
- **ExtendedError**
- **FingerprintPDF**
- **FlateCompression**
- **FontSubsetting**
- **FormName**
- **FullDocumentPath**
- **GeneralFlags**
- **GraphicBitsPerPixel**
- **GraphicsQuality**
- **GrayDownsampleType**
- **GrayFilter**
- **GreyImageFlags**
- **GreyImageResolution**
- **ImageInfoBitsPerPixel**
- **ImageInfoFileType**
- **ImageInfoHeight**
- **ImageInfoTotalPages**
- **ImageInfoWidth**
- **ImageInfoXResolution**
- **ImageInfoYResolution**
- **KeepPortDocument**
- **LastError**
- **LinearizeDocument**
- **MonoFilter**
- **MonoImageFlags**
- **MonoImageResolution**
- **NewDocumentName**
- **NewPortName**
- **NewPrinterName**
- **NewUniqueID**
- **Orientation**

- **OutputDirectory**
- **PaperLength**
- **PaperSize**
- **PaperWidth**
- **PDFAuthor**
- **PDFKeywords**
- **PDFSubject**
- **PDFTimeout**
- **PDFTitle**
- **PoolServerPort**
- **PreserveHalftone**
- **PreserveOverprint**
- **PrinterPoolWaitTimeout**
- **PrintQuality**
- **PrintResolution**
- **ReservedPrinterNumber**
- **Resolution**
- **Scaling**
- **TrueTypeOption**
- **UCRandBGR**
- **UseStaticPool**

ASCIIEncode

Description: ASCII85 encodes binary streams including page content, sampled images and embedded fonts to produce an ASCII PDF. Otherwise the output may contain mostly binary data.

Syntax: `retValue = object.ASCIIEncode = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The ASCIIEncode setting. Available options are: True = Enabled False = Disabled (<i>Default</i>)

retValue:	Value Type	Description & Settings
	BOOLEAN	The current ASCIIEncode setting. Available returns include: True = Enabled False = Disabled

AutoRotate

Description: Automatically rotates pages based on the orientation of the majority of the text flow.

Syntax: `retValue = object.AutoRotate = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The AutoRotate setting. Available options are: True = Enabled (<i>Default</i>) False = Disabled

retValue:	Value Type	Description & Settings
	BOOLEAN	The current AutoRotate setting. Available returns include: True = Enabled False = Disabled

ChunkSize

Deprecated: Use [GetBinaryImage](#) with [DeleteFile](#) instead.

Description: Sets the number of bytes-per-pass for the [PrintToBrowser](#) method.

Syntax: `object.ChunkSize = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	LONG	A value corresponding to the number of bytes-per-pass.

ColorDownsampleType

Description: Samples or downsamples color images to reduce the number of pixels, deleting information. By altering the settings, you can minimize the loss of detail with maximum benefit to the file size.

NOTE: `SetColorDownsampleThreshold` must be set.

Syntax: `retValue = object.ColorDownsampleType = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
setValue	INTEGER	<p>The sample setting. Available options are:</p> <ul style="list-style-type: none"> 0 = No Downsampling: No downsampling occurs. 1 = Average: Downsampling averages the pixel color in a sample group and replaces the entire group with an average of the pixel color at the specified resolution. 2 = Bicubic: Downsampling occurs using a weighted average of the pixel color in a sample group and replaces the entire group with an average of the pixel color at the specified resolution. Bicubic downsampling is more precise, resulting in smoother gradations than Average and Subsample downsampling, yet is also the slowest process. 3 = Subsample: Subsampling chooses a pixel in the center of a sample group and replaces the entire group with that pixel at the specified resolution. Subsampling is the fastest sampling process, but can result in a less smooth and continuous image.

Value Type	Description & Settings
INTEGER	<p>The current sample setting. Available returns include:</p> <ul style="list-style-type: none"> 0 = No Downsampling. 1 = Average. 2 = Bicubic. 3 = Subsample.

ColorFilter

Description: Compresses color images to reduce redundant data producing a smaller PDF size. For color images, compression can have a direct effect on quality.

Syntax: `retValue = object.ColorFilter = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The compression setting. Available options include: <ul style="list-style-type: none"> 0 = No Compression: No compression occurs. 1 = JPEG Compression: JPEG (Joint Photographic Experts Group) encodes images as JPEGs to achieve a smaller size. Typically this is recommended for images exceed the resolution of the printer. 3 = Flate Compression: Flate encodes binary and ASCII image information producing binary images in the PDF.

retValue:	Value Type	Description & Settings
	INTEGER	The current compression setting. Available returns include: <ul style="list-style-type: none"> 0 = No Compression. 1 = JPEG Compression. 3 = Flate Compression.

ColorImageFlags

Deprecated: Use [ColorDownsampleType](#) and [ColorFilter](#) instead.

Description: Sets the [Default Color Image Flags](#) for color images in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.ColorImageFlags = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	LONG	<p>A value corresponding to the Default Color Image Flags. Specify multiple values with an OR statement (<i>&H02 or &H04</i>). Defaults to no sampling, compression or conversion. Available values are:</p> <p>&H01 = Average Downsample</p> <p>Downsample images to ColorImageResolution. Downsampling involves locating similar groups of pixels, averaging their appearance, and then combining them into a single pixel.</p> <p>&H02 = Subsample Downsample</p> <p>Downsamples images to ColorImageResolution. Subsampling involves locating a common pixel in a given sample area and removing the surrounding pixels.</p> <p>&H04 = Flate Compress</p> <p>Compresses images using Flate compression.</p> <p>&H08 = Predictor Compress</p> <p>Compresses images using Flate compression with predictor.</p> <p>&H10 = DCT Low Compress</p> <p>Compresses image by converting it to low compression JPEG.</p> <p>&H20 = DCT Med Compress</p> <p>Compresses image by converting it to medium compression JPEG.</p> <p>&H40 = DCT High Compress</p> <p>Compresses image by converting it to high compression JPEG.</p> <p>&H80 = CMYK2RGB</p> <p>Converts colorspace to RGB when input colorspace is CMYK.</p>

NOTE: You cannot specify two different sampling or compression settings in the same call. These settings might increase or decrease the file size of the output PDF.

Remarks: Overrides the [Default Color Image Flags](#).

ColorImageResolution

Deprecated: Use [SetColorDownsampleThreshold](#) instead.

Description: Sets the resolution for color images in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.ColorImageResolution = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
setValue	FLOAT	The resolution of color images for the output PDF in DPI. In general, 72 to 96 DPI is used for screen output and 300 to 600 is used for printed output.

NOTE: Increasing the resolution will increase the size of the output PDF.

Remarks: Overrides the [Default Color Image Flags Resolution](#) setting.

CompatabilityLevel

Description: Applies the PDF Version level to the output.

Syntax: `retValue = object.CompatabilityLevel = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The PDF Version. Available options are: 0 = PDF Specification 1.3 (<i>Default</i>) 1 = PDF Specification 1.4 2 = PDF Specification 1.5

retValue:	Value Type	Description & Settings
	INTEGER	The PDF Version setting. Available returns include: 0 = PDF Specification 1.3 1 = PDF Specification 1.4 2 = PDF Specification 1.5

ConcatenateToExisting

Description: Appends the output PDF to the PDF identified by the `NewDocumentName` property.

NOTE: The PDFs must be created in immediate succession. Must be used with `NewDocumentName` and called prior to `StartPrinting`.

Syntax: `object.ConcatenateToExisting = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	A value enabling or disabling concatenation. True = Enable concatenation. False = Disable Concatenation. <i>(Default)</i>

Remarks: A new PDF is generated `NewDocumentName` is not used.

DefaultDomain

Description: Overrides the **Domain** for the **ImpersonateUser** method or **DefaultDomain** string value.

Syntax: `object.DefaultDomain = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The domain. Defaults to "." for the local account database.

DefaultPassword

Description: Overrides the **Password** for the **ImpersonateUser** method or **DefaultPassword** string value.

Syntax: `object.DefaultPassword = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The case-sensitive password for the account. Defaults to "".

DefaultPrinter

Description: Specifies the default PostScript® printer to emulate for dynamic printer generation.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.DefaultPrinter = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The exact name of a printer as it appears in your printer dialog.

Remarks: Overrides the [Printer to Emulate](#).

DefaultQueue

Description: Specifies the temporary spooling locating used during PDF generation.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.DefaultQueue = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The full path for the temporary spooling locating used during PDF generation.

Remarks: Overrides the [Queue Directory](#).

DefaultUser

Description: Overrides the **UserID** for the **ImpersonateUser** method or **DefaultUser** string value.

Syntax: `object.DefaultUser = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The case-sensitive user ID for the account. Defaults to "".

DocumentName

Description: Returns the name of the output PDF.

Syntax: `"retValue" = object.DocumentName`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	STRING	The name of the output PDF.

EmbedAllFonts

Description: Embeds all fonts used in the input file.

Syntax: `retValue = object.EmbedAllFonts = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The EmbedAllFonts setting. Available options are: True = Enabled False = Disabled (<i>Default</i>)

retValue:	Value Type	Description & Settings
	BOOLEAN	The current EmbedAllFonts setting. Available returns include: True = Enabled False = Disabled

ExtendedError

Description: Retrieves an extended return code used by activePDF Technical Support.

Syntax: `retValue = object.ExtendedError`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	LONG	<p>An extended value indicating the outcome of the operation.</p> <p>-3 = A Connection Failure.</p> <p>0 = Success.</p> <p># = An extended error has occurred.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

FingerprintPDF

Description: Adds an activePDF Fingerprint to the output PDF. Use in conjunction with the [IsFingerprintValid](#) property to verify integrity.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.FingerprintPDF = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	A value instructing Server to add an activePDF Fingerprint the output PDF. True = Add a Fingerprint. False = Do not add a Fingerprint. <i>(Default)</i>

FlateCompression

Description: Flate encodes binary streams or ASCII text producing a mostly binary PDF, which can decrease the overall output size.

Syntax: `retValue = object.FlateCompression = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The FlateCompression setting. Available options are: True = Enabled (<i>Default</i>) False = Disabled

retValue:	Value Type	Description & Settings
	BOOLEAN	The current FlateCompression setting. Available returns include: True = Enabled False = Disabled

FontSubsetting

Description: Subsets all fonts used in the input file when the percentage of characters used is less than 60%.

Syntax: `retValue = object.FontSubsetting = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The FontSubsetting setting. Available options are: True = Enabled False = Disabled (<i>Default</i>)

retValue:	Value Type	Description & Settings
	BOOLEAN	The current FontSubsetting setting. Available returns include: True = Enabled False = Disabled

FormName

Description: Sets the page size for the output PDF based on the printer's *Form to Tray Assignment*.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.FormName = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	<p>A case-sensitive paper size corresponding to the available sizes for the current printer driver. For example:</p> <ul style="list-style-type: none"> ▶ 11 x 17 ▶ A3 ▶ A4 ▶ Executive ▶ Ledger ▶ Legal ▶ Letter (<i>Default</i>) ▶ Screen ▶ Tabloid <p>NOTE: Refer to the manufacturer's information for available options for each printer driver.</p>

FullDocumentPath

Description: Returns the path and filename of the output PDF.

Syntax: `"retValue" = object.FullDocumentPath`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	STRING	The path and filename for the output PDF.

GeneralFlags

Deprecated: Use [ASCIIEncode](#), [FlateCompression](#), [AutoRotate](#), [PreserveHalftone](#), [PreserveOverprint](#), and [UCRandBGR](#) instead.

Description: Sets the [Default General Flags](#) for the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.GeneralFlags = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	LONG	<p>A value corresponding to the Default General Flags settings. Specify multiple values with an OR statement. (<i>&H02 or &H04</i>). Available values are:</p> <p>&H01 = Mono Thumbnails</p> <p>Any thumbnails added to PDF are monochrome.</p> <p>&H02 = Color Thumbnails</p> <p>Any thumbnails added to PDF are color.</p> <p>&H04 = ASCII Encode</p> <p>Output PDF is suitable for transmission over 7-bit ASCII link. Default is 8-bit binary. (<i>unchecked</i>)</p> <p>&H08 = Flate Compress Text</p> <p>Compresses text and line work with Flate compression.</p> <p>&H10 = Auto Rotate</p> <p>Attempts to rotate the page orientation so that the majority of the text is horizontal.</p> <p>&H20 = Preserve Halftones</p> <p>Preserves the halftone information the PostScript® file.</p> <p>&H40 = Preserve OPI</p> <p>Preserves any OPI 1.3 or 2.0 comments in the PostScript file.</p> <p>&H80 = Preserve Overprint</p> <p>Preserves any overprinting information in the PostScript file.</p> <p>&H100 = Preserve Transfer</p>

Preserves any transfer functions in the PostScript file.

&H200 = **Apply Transfer**

Applies any transfer functions in the PostScript file.

&H400 = **Remove Transfer**

Removes any transfer functions from the PostScript file.

&H800 = **UCR and BGR**

Uses the under-color removal (*UCR*) and black generation (*BGR*) data in the PostScript file.

&H1000 = **BBOX for Page**

Sets the output page size to the input bounding box size.

NOTE: Might prevent clipping with odd-sizes or EPS files.

&H2000 = **BBOX for Crop**

Sets the output crop box size to the input bounding box size.

&H40000 = **Strategy**

Converts device-independent colors to device-dependent colors.

&H80000 = **Emulate Patterns**

Replaces PostScript patterns with Type-3 font.

Remarks: Overrides [Default General Flags](#).

GraphicBitsPerPixel

Description: Defines the number of bits per pixel of the output PDF for the [ImageToPDF](#) method.

Syntax: `object.GraphicBitsPerPixel = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	A value corresponding to the number of bits per pixel. Defaults to 24.

GraphicsQuality

Description: Defines the image quality of the output PDF for the [ImageToPDF](#) method.

Syntax: `object.GraphicsQuality = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	A value from 2 to 255 indicating the output image quality. 2 = Highest quality. 50 = Default quality. 255 = Lowest quality. NOTE: Higher quality might result in larger file sizes.

GrayDownsampleType

Description: Samples or downsamples gray images to reduce the number of pixels, deleting information. By altering the settings, you can minimize the loss of detail with maximum benefit to the file size.

NOTE: `SetGrayDownsampleThreshold` must be set.

Syntax:

```
retValue = object.GrayDownsampleType = setValue
```

Arguments:

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
setValue	INTEGER	<p>The sample setting. Available options are:</p> <p>0 = No Downsampling: No downsampling occurs.</p> <p>1 = Average: Downsampling averages the pixel color in a sample group and replaces the entire group with an average of the pixel color at the specified resolution.</p> <p>2 = Bicubic: Downsampling occurs using a weighted average of the pixel color in a sample group and replaces the entire group with an average of the pixel color at the specified resolution. Bicubic downsampling is more precise, resulting in smoother gradations than Average and Subsample downsampling, yet is also the slowest process.</p> <p>3 = Subsample: Subsampling chooses a pixel in the center of a sample group and replaces the entire group with that pixel at the specified resolution. Subsampling is the fastest sampling process, but can result in a less smooth and continuous image.</p>

retValue:

Value Type	Description & Settings
INTEGER	<p>The current sample setting. Available returns include:</p> <p>0 = No Downsampling.</p> <p>1 = Average.</p> <p>2 = Bicubic.</p> <p>3 = Subsample.</p>

GrayFilter

Description: Compresses gray images to reduce redundant data producing a smaller PDF size. For gray images, compression can have a direct effect on quality.

Syntax: `retValue = object.GrayFilter = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The compression setting. Available options include: <ul style="list-style-type: none"> 0 = No Compression: No compression occurs. 1 = JPEG Compression: JPEG (Joint Photographic Experts Group) encodes images as JPEGs to achieve a smaller size. Typically this is recommended for images exceed the resolution of the printer. 3 = Flate Compression: Flate encodes binary and ASCII image information producing binary images in the PDF.

retValue:	Value Type	Description & Settings
	INTEGER	The current compression setting. Available returns include: <ul style="list-style-type: none"> 0 = No Compression. 1 = JPEG Compression. 3 = Flate Compression.

GreyImageFlags

Deprecated: Use [GrayDownsampleType](#) and [GrayFilter](#) instead.

Description: Sets the *Image Sampling* and/or *Image Compression* settings for grayscale images in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.GreyImageFlags = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	LONG	<p>A value corresponding to the <i>Image Sampling</i> and/or <i>Image Compression</i> settings. Specify multiple values with an OR statement (&H02 or &H04) Defaults to no sampling or compression. Available values are:</p> <p>&H01 = Average Downsample</p> <p>Downsample images to the GreyImageResolution. Downsampling involves locating similar groups of pixels, averaging their appearance, and then combining them into a single pixel.</p> <p>&H02 = Subsample Downsample</p> <p>Downsamples images to the GreyImageResolution. Subsampling involves locating a common pixel in a given sample area and removing the surrounding pixels.</p> <p>&H04 = Flate Compress</p> <p>Compresses images using Flate compression.</p> <p>&H08 = Predictor Compress</p> <p>Compresses images using Flate compression with predictor.</p> <p>&H10 = DCT Low Compress</p> <p>Compresses image by converting it to low compression JPEG.</p> <p>&H20 = DCT Med Compress</p> <p>Compresses image by converting it to medium compression JPEG.</p> <p>&H40 = DCT High Compress</p> <p>Compresses image by converting it to high compression JPEG.</p> <p>NOTE: You cannot specify two different sampling or compression settings in the same call. These settings might increase or decrease the file size of the output PDF.</p>

Remarks: Overrides [Default Grey Image Flags](#).

GreyImageResolution

Deprecated: Use [SetGrayDownsampleThreshold](#) instead.

Description: Sets the resolution for grayscale images in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.GreyImageResolution = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
setValue	FLOAT	The resolution of grayscale images for the output PDF in DPI. In general, 72 to 96 DPI is used for screen output and 300 to 600 is used for printed output.

NOTE: Increasing the resolution will increase the size of the output PDF.

Remarks: Overrides [Default Grey Image Flags Resolution](#) setting.

ImageInfoBitsPerPixel

Description: Returns the number of bits per pixel in the image specified with [ImageToPDF](#) after calling [LoadImageInfo](#).

Syntax: `retValue = object.ImageInfoBitsPerPixel`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	The number of bits per pixel for the image.

ImageInfoFileType

Description: Returns the file type of the image specified with [ImageToPDF](#) after calling [LoadImageInfo](#).

Syntax: `retValue = object.ImageInfoFileType`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	<p>A value corresponding to the image type.</p> <p>1 = Microsoft® Windows® Bitmap (BMP), OS/2™ Bitmap (BMP), Microsoft® Windows® Device Independent Bitmap (DIB), Portable Bitmap (PBM), Portable GrayMap (PGM), Portable PixelMap (PPM), NeoPaint® for MS-DOS® thumbnails (ST), NeoPaint® for Windows® thumbnails (STW), Raster Sun Microsystems™ (SUN), PC PaintBrush® (PCX).® Windows® Bitmap (BMP)."</p> <p>2 = PC PaintBrush® (PCX).</p> <p>3 = TIFF w/o compression, RGB color space, 8-bit grayscale (TIF).</p> <p>4 = TrueVision® TARGA® (TGA).</p> <p>5 = Microsoft® Windows® Icons (ICO), OS/2™ Icons (ICO).</p> <p>6 = Dr. Halo®, Dr. Genius clipboard Format (CUT).</p> <p>7 = GEM IMG (IMG), IMG Software Set (IMG), Vivid Raytracer (IMG).</p> <p>8 = CMYK Separated Images in TIFF 6.0 file format (SEP), Tagged Interchange File Format (TIFF).</p> <p>9 = Compressed Microsoft® Windows® BMP (RLE), Compressed Images in SGI® File Format (RLE), Compressed Intergraph Raster images (RLE), Utah Run Length Encoded (RLE).RegisteredTrademark> Windows® BMP (RLE),</p> <p>10 = Adobe® Photoshop® 2.5 - 4.0 (PSD).</p> <p>12 = Raster Sun Microsystems™ (RAS).</p> <p>13 = Macintosh® PICT (PCT).</p> <p>14 = Amiga™ Interchange File Format (LBM).</p> <p>15 = JPEG File Format (JFF), JPEG File Interchange Format (JIF), JPEG (JPEG).</p>

16 = Microsoft® Paint (*MSP*).

17 = RIPTerm™ images (*ICN*).

18 = Tetris™ Images (*JMX*).

19 = Slow Scan Television (*HRZ*).

20 = Autodesk® Animator™ Pro animation (*FLC*), Autodesk® Animator™ animation (*FLI*).

21 = CALS Raster Type-I (*CAL*), CALS Raster Type-I(*MIL*).

22 = SGI® black & white image files (*BW*), SGI® TrueColor™ RGB images (*RGB*), SGI® Image File Format (*SGI*).

23 = Portable Network Graphics (*PNG*).

24 = TIFF w/ JPEG compression, YUV 4:2:2 color space (*TIF*).

25 = DEGAS Elite(*PC2*).

26 = MacPaint® (*MAC*).

27 = Microsoft® Windows® cursors (*CUR*).

28 = CorelDraw® 2.0-9.0 previews and imported pics (*CDR*), CorelDraw® 2.0-9.0 Templates (*CDT*), Corel® metafile exchange 5.0-9.0 preview and imported bitmaps only (*CMX*), Corel® PhotoPaint® 6.0 (*CPT*).

29 = X Windows™ file (*X*).

31 = Portable Any Bitmap (*PNM*).

32 = Intel™ multi-page fax format (*DCX*).

33 = WordPerfect® graphics (*WPG*).

34 = PC Paint, Pictor images (*PIC*), Dr. Halo®, Dr. Genius picture format (*PIC*).

36 = AMIPro® text document images (*SAM*).

37 = Autodesk® Animator™ Pro animation(*CEL*), Autodesk® Animator™ still picture (*CEL*).

38 = Zoner Zebra for Windows® 1.0 - 1.5 (*ZBR*).

39 = UDI (*UDI*).

40 = Microsoft® Windows® clipboard file (*CLP*).

42 = OLIFAX fax package (*OFX*).

43 = Kodak™ PhotoCD™ (*PCD*).

44 = Epson® scanner format (*PYX*).

45 = Zoner Bitmap (*BMI*).

46 = Zoner metafile (*ZMF*).

47 = Microsoft® at Work™ Fax (*AWD*).

48 = Quick Link II™ fax format (*QFX*).

50 = CALS Raster (*CAL*).

51 = Windows® metafile (*WMF*).

53 = Microsoft Paint (*MSP*).

54 = WordPerfect (*WPD*).

55 = Raster Sun Microsystems (*RAS*).

56 = MacPict (*PCT*).

57 = Kodak PhotoCD (*PCD*).

66 = Raw FAX w/ CCITT group 3, 1-D (*FAX*).

67 = Raw FAX w/ CCITT group 3, 2-D (*FAX*).

68 = Raw FAX w/ CCITT group 4 (*FAX*).

69 = Winfax w/ CCITT group 3, 1-D (*FXS*).

70 = Winfax w/ CCITT group 4 (*FXS*).

71 = TIFF, IOCA w/ CCITT group 3, 1-D (*TIFF*).

72 = TIFF, IOCA w/ CCITT group 3, 2-D (*TIFF*).

73 = TIFF, IOCA w/ CCITT group 4 (*TIFF*).

74 = OS/2 BMP version 2.x (*BMP*).

75 = Portable Network Graphics (*PNG*).

76 = Adobe Photoshop 3.0 (*PSD*).

77 = TIFF, IOCA w/ CCITT group 3, 1-D, w/o MO:DCA (*TIFF*).

78 = TIFF, IOCA w/ CCITT group 3, 2-D, w/o MO:DCA (*TIFF*).

79 = TIFF, IOCA w/ CCITT group 4, w/o MO:DCA (*TIFF*).

84 = Windows Bitmap w/ RLE compression (*BMP*).

85 = TIFF w/o compression, CMYK color space (*TIF*).

87 = TIFF w/ PackBits, RGB color space (*TIF*).

88 = TIFF w/ PackBits, CMYK color space (*TIF*).

93 = TIFF w/o compression, YCbCr color space (*TIF*).

95 = TIFF w/ PackBits, YCbCr color space (*TIF*).

96 = Exif TIFF image, w/o compression, RGB color space (*EXF*).

97 = Exif TIFF image, w/o compression, YCbCr color space (*EXF*).

98 = Exif JPEG image, YUV 4:2:2 color space (*EXF*).

99 = Microsoft FAX - Win32 (*MSF*).

101 = Exif JPEG image, YUV 4:1:1 color space (*EXF*).

102 = Portable Bitmap - ASCII (*PBF*).

103 = Portable Bitmap - Binary (*PBF*).

104 = Portable Greymap - ASCII File (*PGM*).

105 = Portable Greymap - Binary File (*PGM*).

106 = Portable Pixelmap - ASCII File (*PGM*).

107 = Portable Pixelmap - Binary File (*PGM*).

108 = Dr. Halo (*CUT*).

109 = XPicMap (*XPM*).

110 = XBitmap (*XMP*).

111 = Encapsulated PostScript Interchange Format (*EPI*).

112 = Encapsulated PostScript Interchange Format Interchange - Multi (*EPI*).

114 = Windows Clipboard (*CLP*).

116 = Windows Enhanced Meta File (*EMF*).

129 = CALS CCITT G4 Tiled (*CAL*).

130 = CALS CCITT G4 (*CAL*).

131 = CALS Fax Group 4(*CAL*).

132 = X Window Dump - v.10 (*X10*).

133 = X Window Dump - v.11 (*X11*).

ImageInfoHeight

Description: Returns the height in pixels for the image specified with [ImageToPDF](#) after calling [LoadImageInfo](#).

Syntax: `retValue = object.ImageInfoHeight`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	The height of the image in pixels.

ImageInfoTotalPages

Description: Returns the total number of pages for the image file specified with [ImageToPDF](#) after calling [LoadImageInfo](#).

Syntax: `retValue = object.ImageInfoTotalPages`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	The total number of pages in the image file.

ImageInfoWidth

Description: Returns the width in pixels for the image specified with [ImageToPDF](#) after calling [LoadImageInfo](#).

Syntax: `retValue = object.ImageInfoWidth`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	The width of the image in pixels.

ImageInfoXResolution

Description: Returns the horizontal resolution in Dots Per Inch (*DPI*) for the image specified with [ImageToPDF](#) after calling [LoadImageInfo](#).

Syntax: `retValue = object.ImageInfoXResolution`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	The horizontal resolution of the image in DPI.

ImageInfoYResolution

Description: Returns the vertical resolution in Dots Per Inch (*DPI*) for the image specified with [ImageToPDF](#) after calling [LoadImageInfo](#).

Syntax: `retValue = object.ImageInfoYResolution`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	INTEGER	The vertical resolution of the image in DPI.

KeepPortDocument

Description: Instructs Server to retain the temporary PostScript® file created during PDF generation. This document can be used for diagnostic purposes.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.KeepPortDocument = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	A value instructing Server to keep the temporary PostScript file. True = Keep PostScript file. False = Do not keep PostScript file. <i>(Default)</i>

Remarks: The file is retained in the X:\WINDOWS\activePDF or X:\WINNT\activePDF directory depending on the installed operating system. The file will have a name similar to *activePDF-22.PS*.

NOTE: Additional application specific settings might be required when using activePDF DocConverter™ or WebGrabber™.

LastError

Description: Retrieves a return code used by activePDF Technical Support.

Syntax: `retValue = object.LastError`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	LONG	<p>A value indicating the outcome of the operation.</p> <p>+# = A Windows® error.</p> <p>0 = Success.</p> <p>-# = An activePDF Server error.</p> <p>NOTE: Contact Technical Support if 0 is not returned</p>

LinearizeDocument

Description: Enables linearization (*byte-serving*) for the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.LinearizeDocument = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	A value instructing Server to linearize the output PDF. True = Output is linearized. False = Output is not linearized. (<i>Default</i>)

MonoFilter

Description: Compresses monochrome images to reduce redundant data producing a smaller PDF size. For monochrome images, compression can have a direct effect on quality.

Syntax: `retValue = object.MonoFilter = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	<p>The compression setting. Available options include:</p> <ul style="list-style-type: none"> 0 = No Compression: No compression occurs. 1 = CCITT Compression: CCITT (International Coordinating Committee for Telephony and Telegraphy) divides the original image into scan lines and then compresses each line based on the previous line. 2 = Flate Compression: Flate encodes binary and ASCII image information producing binary images in the PDF. 3 = RunLength Compression: RunLength stores consecutive data segments containing identical information as a single data to reduce redundant information. Images containing large areas of a single color will experience the greatest results with this compression.

retValue:	Value Type	Description & Settings
	INTEGER	<p>The current compression setting. Available returns include:</p> <ul style="list-style-type: none"> 0 = No Compression. 1 = CCITT Compression. 2 = Flate Compression. 3 = RunLength Compression.

MonoImageFlags

Deprecated: Use [MonoDownsampleType](#) and [MonoFilter](#) instead.

Description: Sets the *Image Sampling* and/or *Image Compression* settings for monochrome images in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.MonoImageFlags = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	LONG	<p>A value corresponding to the <i>Image Sampling</i> and/or <i>Image Compression</i> settings. Specify multiple values with an OR statement (<i>&H02 or &H04</i>). Defaults to no sampling or compression. Available values are:</p> <p>&H02 = Subsample Downsample</p> <p>Downsamples images to the MonoImageResolution. Subsampling involves locating a common pixel in a given sample area and removing the surrounding pixels.</p> <p>&H04 = Flate Compress</p> <p>Compresses image using Flate compression.</p> <p>&H08 = Predictor Compress</p> <p>Compresses image using Flate compression with a predictor.</p> <p>&H10 = CCITT Compress</p> <p>Compresses image using CCITT group 4-fax compression.</p> <p>NOTE: You cannot specify two different compression settings in the same call. These settings might increase or decrease the file size of the output PDF.</p>

Remarks: Overrides the [Default Monochrome Image Flags](#).

MonoImageResolution

Deprecated: Use [SetMonoDownsampleThreshold](#) instead.

Description: Sets the resolution for monochrome images in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.MonoImageResolution = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
setValue	FLOAT	The resolution of monochrome images for the output PDF in DPI. In general, 72 to 96 DPI is used for screen output and 300 to 600 is used for printed output.

NOTE: Increasing the resolution will increase the size of the output PDF.

Remarks: Overrides the [Default Monochrome Image Flags Resolution](#) setting.

NewDocumentName

Description: Sets the filename for the output PDF. By default, Server uses an auto-generated unique identifier.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.NewDocumentName = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The filename for the output PDF.

NOTE: Use the [OutputDirectory](#) property to set the file path.

Remarks: One call is required for each file generated or the new file overwrites the previous file.

NewPortName

Description: Returns the path to the PostScript® generated as part of the PDF generation process.

NOTE: Must be called after [StopPrinting](#)

Syntax: `"retValue" = object.NewPortName`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	STRING	The path to the PS file.

NewPrinterName

Description: Returns the name of the virtual printer created by Server for use in PDF generation

NOTE: Must be called after [StopPrinting](#)

Syntax: `"retValue" = object.NewPrinterName`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	STRING	The name of the virtual printer.

NewUniqueID

Description: Returns the unique identifier created by Server for use in PDF generation

NOTE: Must call after [StopPrinting](#).

Syntax: `"retValue" = object.NewUniqueID`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.

retValue:	Value Type	Description & Settings
	STRING	The unique identifier.

Remarks: By default, the unique identifier is applied as the output filename unless [NewDocumentName](#) is called. You must call the [FromString](#) method first when using activePDF WebGrabber™.

Orientation

Description: Sets the printer *PageOrientation* for the output PDF.

NOTE: Must call prior to [StartPrinting](#).

Syntax: `object.Orientation = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	A value indicating Portrait or Landscape orientation. 1 = Portrait. 2 = Landscape.

OutputDirectory

Description: Specifies the directory where activePDF Server will place the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.OutputDirectory = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The directory to place the output PDF. Defaults to the Default Output Path . Overrides when set.

PaperLength

Description: Sets the *Length* under *Custom Page Size Dimensions* for the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PaperLength = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The page length in PDF Units . Defaults to 612 PDF Units .

PaperSize

Description: Sets the *Paper Size* for the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PaperSize = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	A value corresponding to the page size of the output PDF. 1 = Letter (8 1/2 x 11 in.) 2 = Letter Small (8 1/2 x 11 in.) 3 = Tabloid (11 x 17 in.) 4 = Ledger (17 x 11 in.) 5 = Legal (8 1/2 x 14 in.) 6 = Statement (5 1/2 x 8 1/2 in.) 7 = Executive (7 1/2 x 10 1/2 in.) 8 = A3 (297 x 420 mm) 9 = A4 (210 x 297 mm) 10 = A4 Small (210 x 297 mm) 11 = A5 (148 x 210 mm) 12 = B4 (250 x 354 mm) 13 = B5 (182 x 257 mm) 14 = Folio (8 1/2 x 13 in.) 15 = Quarto (215 x 275 mm) 16 = Bristol (10 x 14 in.) 17 = Tabloid (11 x 17 in.) 18 = Note (8 1/2 x 11 in.) 19 = Envelope #9 (3 7/8 x 8 7/8 in.) 20 = Envelope #10 (4 1/8 x 9 1/2 in.) 21 = Envelope #11 (4 1/2 x 10 3/8 in.)

- 22 = Envelope #12 ($4\frac{1}{2} \times 11$ in.)
- 23 = Envelope #14 ($5 \times 11\frac{1}{2}$ in.)
- 24 = Architectural C (18×24 in.)
- 25 = Architectural D (24×36 in.)
- 26 = Architectural E (36×48 in.)
- 27 = Envelope DL (110×220 mm)
- 28 = Envelope C3 (324×458 mm)
- 29 = Envelope C4 (229×324 mm)
- 30 = Envelope C5 (162×229 mm)
- 31 = Envelope C6 (114×162 mm)
- 32 = Envelope C65 (114×229 mm)
- 33 = Envelope B4 (250×353 mm)
- 34 = Envelope B5 (176×250 mm)
- 35 = Envelope B6 (176×125 mm)
- 36 = Envelope (110×230 mm)
- 37 = Envelope Monarch ($3\frac{7}{8} \times 7\frac{1}{2}$ in.)
- 38 = Envelope ($3\frac{5}{8} \times 6\frac{1}{2}$ in.)
- 39 = U.S. Standard Fanfold ($14\frac{7}{8} \times 11$ in.)
- 40 = German Standard Fanfold ($8\frac{1}{2} \times 12$ in.)
- 41 = German Legal Fanfold ($8\frac{1}{2} \times 13$ in.)

NOTE: Certain printers might not support all page sizes.

PaperWidth

Description: Sets the *Width* under *Custom Page Size Dimensions* for the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PaperWidth = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The page width in PDF Units . Defaults to 612 PDF Units .

PDFAuthor

Description: Defines the value of the *Author* setting in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PDFAuthor = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The desired author for the output PDF.

PDFKeywords

Description: Defines the value of the *Keywords* setting in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PDFKeywords = "setValue"`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
setValue	STRING	The desired keywords for the output PDF. Separate multiple keywords with commas followed by no-space.

PDFSubject

Description: Defines the value of the *Subject* setting in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PDFSubject = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The desired subject for the output PDF.

PDFTimeout

Description: Sets the wait time in seconds for the output PDF to be generated. Used with activePDF WebGrabber™.

Syntax: `object.PDFTimeout = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The wait time in seconds. Default is 30.

PDFTitle

Description: Defines the value of the *Title* setting in the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PDFTitle = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	STRING	The desired title for the output PDF.

PoolServerPort

Description: Defines the port used by activePDF Server. The port must remain open and dedicated to Server.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PoolServerPort = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The open and dedicated port used by Server. Default is 53535.

Remarks: Overrides the [Server I/P Port](#).

PreserveHalftone

Description: Retains any existing halftone information in files being converted to PDF. Halftone information is intended for use with specific output devices.

Syntax: `retValue = object.PreserveHalftone = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The PreserveHalftone setting. Available options are: True = Enabled False = Disabled (<i>Default</i>)

retValue:	Value Type	Description & Settings
	BOOLEAN	The current PreserveHalftone setting. Available returns include: True = Enabled False = Disabled

PreserveOverprint

Description: Retains any existing overprint settings in files being converted to PDF. Overprint exists when multiple inks print on top of each other to produce a combined color. Failing to preserve the overprint results in only one color being printed. Overprint information is intended for use with specific output devices.

Syntax: `retValue = object.PreserveOverprint = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The PreserveOverprint setting. Available options are: True = Enabled False = Disabled (<i>Default</i>)

retValue:	Value Type	Description & Settings
	BOOLEAN	The current PreserveOverprint setting. Available returns include: True = Enabled False = Disabled

PrinterPoolWaitTimeout

Deprecated: Function no longer required in updated Server process.

Description: Sets the wait time in seconds before Server pulls a printer from the [Static](#) or [Reserved Pool](#).

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PrinterPoolWaitTimeout = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	SHORT	The wait time in seconds. Defaults to 30.

PrintQuality

Deprecated: Use [PrintResolution](#) instead.

Description: Overrides the printer's default *Print Quality* setting for the output PDF.

NOTE: Must call prior to [StartPrinting](#).

Syntax: `object.PrintQuality = setValue`

Argument	Value Type	Description & Settings
object	N/A	Set to an instance of the Server object.
setValue	INTEGER	The resolution for the output PDF in DPI or one of the available preconfigured settings. In general, 72 to 96 DPI is used for screen output and 300 to 600 is used for printed output.

NOTE: Increasing the resolution will increase the size of the output PDF.

Available preconfigured settings:

-1 = **Draft**

-2 = **Low**

-3 = **Medium**

-4 = **High**

PrintResolution

Description: Overrides the printer's default *Print Quality* setting of the *Print Driver* assigned to the printer.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.PrintResolution = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	INTEGER	The resolution for the output PDF in DPI. In general, 72 to 96 DPI is used for screen output and 300 to 600 is used for printed output. NOTE: Increasing the resolution will increase the size of the output PDF.

ReservedPrinterNumber

Deprecated: Function no longer required in updated Server process.

Description: Instructs Server to use a specific printer number from the [Reserved Pool](#) during conversion.

NOTE: Must be called prior to [StartPrinting](#). You must create the [Reserved Pool](#) before conversion.

Syntax: `object.ReservedPrinterNumber = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	SHORT	A number corresponding to a Reserved Printer to use during conversion.

Resolution

Description: Sets the resolution of the output PDF in DPI.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.Resolution = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	FLOAT	The resolution for the output PDF in DPI. In general, 72 to 96 DPI is used for screen output and 300 to 600 is used for printed output. NOTE: Increasing the resolution will increase the size of the output PDF.

Remarks: Printed quality depends on the printer's resolution. If set higher than 600, thumbnails are not embedded in the output PDF. Certain PostScript® operations change elements to machine pixels, which might produce undesirable results when the PDF and printer resolution do not match. This is also true if the machine and printer resolution are different.

Scaling

Description: Overrides the printer's default *Scaling* setting for the output PDF.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.scaling = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	SHORT	A value corresponding to the percentage by which the rendered size of the output PDF is increased or decreased.

TrueTypeOption

Description: Overrides the selected printer's default *TrueType*® *Font Download Option*.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.TrueTypeOption = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	SHORT	<p>A value corresponding to the TrueType setting. Available values are:</p> <p>1 = Automatic (<i>Default</i>)</p> <p><i>The PostScript driver decides the format.</i></p> <p>2 = Outline</p> <p><i>Downloads TrueType fonts as scalable outline fonts.</i></p> <p>3 = Native TrueType</p> <p><i>For TrueType rasterizer printers, downloads as outline font.</i></p>

UCRandBGR

Description: Undercolor Removal (UCR) lowers the amount of Cyan, Magenta, and Yellow to compensate for the Black Generation (BGR). BGR is the amount of black necessary to reproduce a particular color. UCR and BGR information is intended for use with specific output devices.

Syntax: `"retValue" = object.UCRandBGR = "setValue"`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	The UCRandBGR setting. Available options are: /Preserve = Retains any existing settings in files being converted to PDF. <i>(Default)</i> /Remove = Removes any existing settings.

retValue:	Value Type	Description & Settings
	BOOLEAN	The current UCRandBGR setting. Available returns include: /Preserve = Retains any existing settings in files being converted to PDF. /Remove = Removes any existing settings.

UseStaticPool

Deprecated: Function no longer required in updated Server process.

Description: Enables the use of the [Static Pool](#) during conversion.

NOTE: Must be called prior to [StartPrinting](#).

Syntax: `object.UseStaticPool = setValue`

Arguments:	Argument	Value Type	Description & Settings
	object	N/A	Set to an instance of the Server object.
	setValue	BOOLEAN	A value enabling or disabling the use of the Static Pool printers. True = Enable Static Pool. False = Disable Static Pool. (<i>Default</i>)

Advanced Configuration

Server includes several advanced techniques designed to expand the possibilities of your implementation. This section details these techniques, including step by step instructions and examples.

NOTE: We strongly recommend that you familiarize yourself with the basic operations of Server before implementing advanced configurations.

- ▶ **Using Server in .NET**
- ▶ **Using Registry Switches**
- ▶ **activePDF FileCleaner Utility**

Using Server in .NET

The following instructions are for implementing activePDF Server in a .NET environment using the `APServerNET.DLL COM Interop` wrapper.

Installing the APServerNET.DLL Wrapper

The `APServerNET.DLL COM Interop` wrapper is not installed with activePDF Server. After installing Server, you need to download the wrapper from http://www.activePDF.com/altdownloads/Products/dotNET/APServerNET_SP7.zip, and then unzip the DLL to a location accessible by your .NET scripting software.

Namespace in .NET

In .NET, the Server namespace becomes `APServerNET.APServer`.

Referencing the APServerNET.DLL

You must add a reference to the `APServerNET.DLL` in your .NET scripting environment. A missing or invalid reference will result in a *missing type* or *namespace* error. Use the following steps to add a reference in Visual Studio® .NET:

1. Start Visual Studio .NET.
2. In the **Solution Explorer**, right-click **Reference**, and then select **Add Reference**.
3. In the **Add Reference** dialog, click **Browse**.
4. In the **Browse** field, type the location of the `APServerNET.DLL`, and then click **Open**.

Instantiating the Object

After referencing the .NET DLL, you must instantiate the Server object in .NET using the syntax of the script language. Instantiation instructions for VB.NET and C# are provided below.

VB.NET

1. Add the following line to the beginning of your .NET code:

```
Imports APServerNET
```

2. Instantiate the Server object:

```
'Instantiate the Server object.  
Dim SVR As APServerNET.APServer  
SVR = New APServer
```

C#

1. Add the following line to the beginning of your .NET code:

```
using APServerNET
```

2. Instantiate the Server object:

```
//Instantiate the Server object.  
APServerNET.APServer SVR = new APServer();
```

Using Registry Switches

The following `HKEY_LOCAL_MACHINE\SOFTWARE\activePDFServer` registry key string values are listed for legacy purposes only. These switches can set in the [Configuration Editor](#). Do not alter or create these values unless instructed by [Technical Support](#).

CAUTION: We strongly recommend that you back up the registry before making any changes. Incorrect changes to the registry may result in permanent data loss or damaged files. Make sure that you modify only the keys specified. For additional information on editing the registry, refer to you Microsoft Windows documentation before proceeding.

String Value	Description & Settings
<code>APXLocation</code>	Use the APX Location option.
<code>CPUThreshold</code>	Use the CPU Threshold setting.
<code>DebugMode</code>	Use the Debug Mode option.
<code>DefaultColorImageFlags</code>	Use the Default Color Image Flags settings.
<code>DefaultColorResolution</code>	Use the Default Color Image Flags settings.
<code>DefaultDomain</code>	Sets the Domain for the ImpersonateUser method. Can be used to hide the domain from the script.
<code>DefaultGeneralFlags</code>	Use the Default General Image Flags settings.
<code>DefaultGreyImageFlags</code>	Use the Default Grey Image Flags settings.
<code>DefaultGreyResolution</code>	Use the Default Grey Image Flags settings.
<code>DefaultPDFDirectory</code>	Use the Default Output Path setting.
<code>DefaultPDFExtension</code>	Use the PDF Extension setting.
<code>DefaultPrinter</code>	Use the Default Printer To Emulate setting.
<code>DefaultQueue</code>	Use the Default queue location setting.
<code>DefaultPassword</code>	Sets the Password for the ImpersonateUser method. Can be used to hide the password from the script.
<code>DefaultResolution</code>	Use the Whole File Resolution setting.
<code>DefaultUser</code>	Sets the UserID for the ImpersonateUser method. Can be used to hide the user from the script.
<code>DisablePriorityBoosting</code>	Use the Disable priority boosting option.
<code>DisableSubset</code>	Use the Disable font subsetting option.
<code>EmbedBase14</code>	Use the Embed Base 14 fonts option.
<code>EnableWatcher</code>	Use the Enable crash watch thread option.
<code>GenerateUniqueTempDirs</code>	Instructs Server to generate unique temporary queues for each conversion. This might prevent latency issues using multiple instances of Server to convert graphic intensive input. <ul style="list-style-type: none"> ▶ 0 = Disabled (<i>Default</i>) ▶ 1 = Enabled
<code>IgnoreNotDefFix</code>	Use the Ignore NOTDEF Fix option.

InMemoryAPX	Use the In Memory APX Files option.
Installed	Indicates the product OCX is installed.
JobPeekMinutes	Use the Job Peek Minutes setting.
Linearize	Use the Linearize upon completion option.
MaxPoolSize	Use the Standard Pool Size setting.
MaxReservedPoolSize	Use the Reserved Pool Size setting.
MaxStaticPoolSize	Use the Static Pool Size setting.
PDFBufferSize	Use the Conversion buffer size setting.
PoolServerPort	Use the Server I/P Port setting.
PrintDirect	Use the Enable PrintDirect flag option.
ProcessPriority	Use the Process Priority setting.
SingleThreadedPrinter	Use the Single threaded printer name setting.
SubstituteNoFonts	Use the Substitute no fonts option.
VMSize	Use the Override VM size setting.
WindowsDirectory	Use the Windows directory setting.

activePDF FileCleaner Utility

The *activePDF FileCleaner Utility* enables you to schedule file deletion for a specified directory based on the age of the file. This can be useful for preventing temporary or duplicate file build up in a directory used as part of your activePDF Solution.

NOTE: File cleaning will take place at regular intervals based on the file age. At this time, there is no option for specifying how often the file cleaning will occur. The *FileCleaner Utility* is also installed when installing activePDF Server.

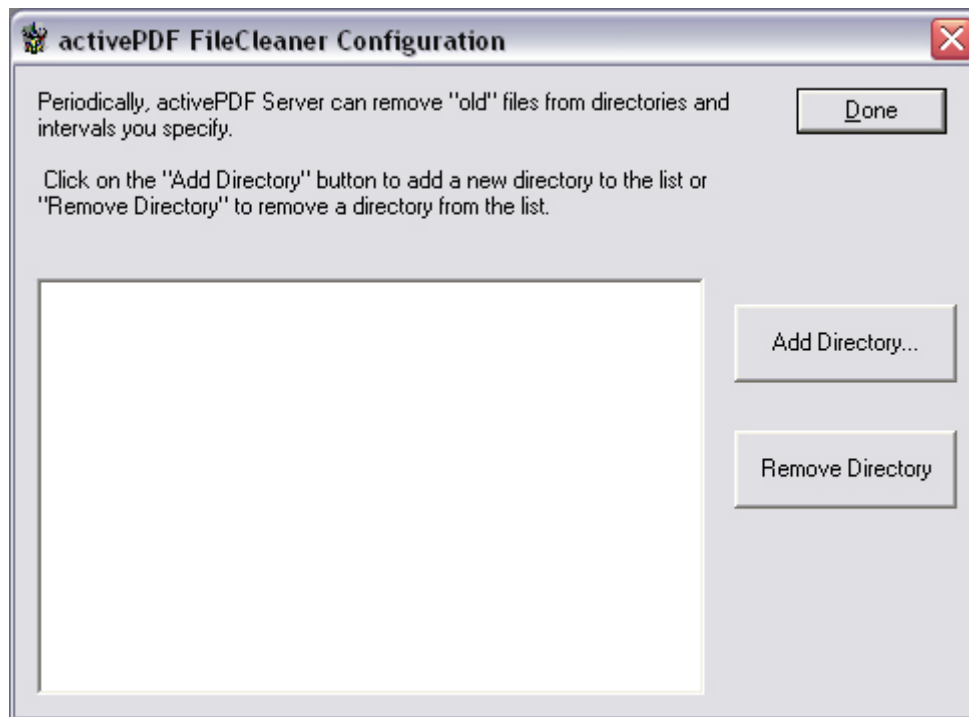


Figure 6.1: The activePDF FileCleaner Configuration Dialog.

The FileCleaner Utility has these options:

- ▶ Add Directory
- ▶ RemoveDirectory
- ▶ Done

Add Directory

Add Directory is used to specify the directory for file cleaning.

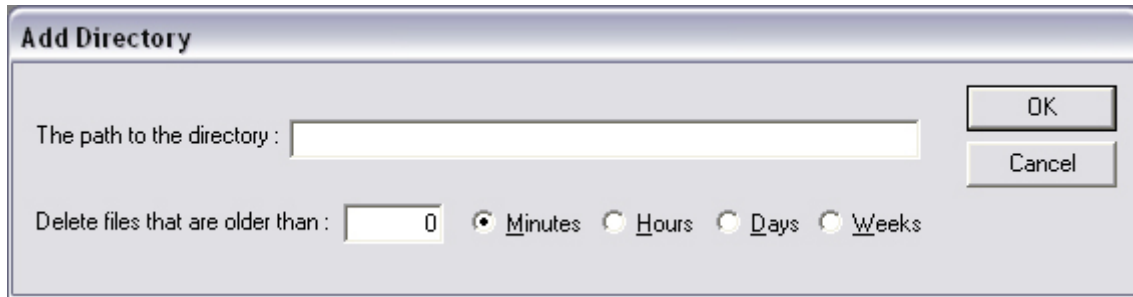


Figure 6.2: The Add Directory Dialog.

Option	Description
The path to the directory	The path to the local directory where the file cleaning will take place.
Delete Files that are older than	<p>The age of the files to delete, specified in Minutes, Hours, Days or Weeks. After typing a number, you will need to specify the unit of time by clicking to select one of the adjacent time units. For example, if you specify 10 Minutes, the <i>FileCleaner Utility</i> will remove all files older than 10 minutes.</p> <p>NOTE: By default, this is set to 0 (zero) minutes. If this option is left unchanged or set to zero, all files will be deleted from the specified directory.</p>
OK	<p>Adds the new directory and settings.</p> <p>CAUTION: File cleaning will take place once you click OK. We strongly recommend that you ensure the path to the local directory is correct, and you have backed up all necessary files. There is no way to edit a directory once added. If you need to change the directory, you will need to use the <i>Remove Directory</i> option.</p>
Cancel	Exit and quit without saving changes.

RemoveDirectory

Remove the selected directory from the list. The directory will not be cleaned again until added.

Done

Close the *FileCleaner Utility* and save all changes.

Appendices

Refer to the following appendices for additional data regarding Server:

- ▶ **PDF Coordinates and Units**
- ▶ **PDF Views**
- ▶ **PDFMarks**
- ▶ **Development Examples**
- ▶ **Server Updates**

PDF Coordinates and Units

The x-axis extends to the right from the page origin, and the y-axis extends upward from the page origin. The unit of measurement is given as *PDF Units* where 72 *PDF Units* equal 1 *inch*. This means that the default 8.5 in. x 11 in. page is equal to 612 by 792 *PDF Units*. This measurement is found by multiplying the two dimensions by 72.

PDF Views

PDF views control the display of the output PDF when it is first opened or a display link is followed. They are useful when you need to present a PDF for a specific purpose and to guide the user's attention to key points. The following PDF View strings are available:

- ▶ Fit
- ▶ FitB
- ▶ FitBH
- ▶ FitBV
- ▶ FitH
- ▶ FitR
- ▶ FitV
- ▶ XYZ

Fit

Description: Sets the size of the display dialog to wrap around the PDF's crop box dimensions.

Syntax: `"/View [/Fit]"`

FitB

Description: Sets the size of the display dialog to wrap around the PDF's bounding box dimensions.

Syntax: `"/View [/FitB]"`

FitBH

Description: Sets the size of the display dialog to wrap around the PDF's bounding box dimensions.

Syntax: `"/View [/FitBH top]"`

Parameters:

Parameter	Description & Settings
top	The vertical offset distance from the page-origin to the top edge of the screen window in default user space. Set to <code>NULL</code> to retain the current value.

FitBV

Description: Sets the size of the display dialog to wrap around the PDF's bounding box dimensions.

Syntax: `"/View [/FitBV left]"`

Parameters:

Parameter	Description & Settings
left	The horizontal offset distance from the page-origin to the left edge of the screen window in default user space. Set to <code>NULL</code> to retain the current value.

FitH

Description: Sets the size of the display dialog to fit the PDF's width.

Syntax: `"/View [/FitH top]"`

Parameters:

Parameter	Description & Settings
top	The vertical offset distance from the page-origin to the top edge of the screen window in default user space. Set to <code>NULL</code> to retain the current value.

FitR

Description: Sets the size and location of the display window to the location specified in default user space.

Syntax: `"/View [/FitR x1 y1 x2 y2]"`

Parameter	Description & Settings
x1	The offset distance from the page origin to the left edge of the screen window in default user space.
y1	The offset distance from the page origin to the bottom edge of the screen window in default user space.
x2	The horizontal offset distance from the page origin to the upper-right corner of the page in default user space.
y2	The vertical offset distance from the page origin to the upper-right corner of the page in default user space.

FitV

Description: Sets the size of the display dialog to fit the PDF's height.

Syntax: `"/View [/FitV left]"`

Parameter	Description & Settings
left	The horizontal offset distance from the page-origin to the left edge of the screen window in default user space. Set to <code>NULL</code> to retain the current value.

XYZ

Description: Sets the origin-offset destination and zoom that is displayed when the link is followed.

Syntax:

```
"/View [/XYZ left top zoom]"
```

Parameters:

Parameter	Description & Settings
left	The horizontal offset distance from the page-origin to the center of the display area in default user space. Set to <code>NULL</code> to retain the current value of the open PDF.
top	The vertical offset distance from the page-origin to the cent of the display area in default user space. Set to <code>NULL</code> to retain the current value of the open PDF.
zoom	The magnification factor used to display the area when the link is followed. 1 = 100% 0 = Same as <code>NULL</code>

PDFMarks

PDFMarks are PostScript® code representing PDF features such as Bookmarks, Form Fields, and Comments. The PDFMarks are added to the document PostScript and generated when converted to PDF.

Types of PDFMarks

The following are common PDFMarks:

- ▶ **/ANN** - Annotations
- ▶ **/OUT** - Bookmarks
- ▶ **/ARTICLE** - Articles
- ▶ **/PAGES /PAGE** - Page Cropping
- ▶ **/DOCINFO** - Info Dictionary
- ▶ **/DOCVIEW** - Document Open Options
- ▶ **/PAGELABEL** - Page Label and Plate Color

PDFMarks require a firm understanding of the PostScript and PDF structure to implement. Once understood, PDFMarks can be used to accomplish many tasks quicker and are the only way to accomplish certain PDF features.

Using PDFMarks

The PDFmark structure has three required parts, known as the operator. The three parts are (*Example given*):

- ▶ **Mark Object** - Denoted by the [character.

```
[
```

- ▶ **Arguments** - Details the features of the PDFmark.

```
/Rect [ 75 586 456 663 ]  
/Contents (Note Type Comment Example.)
```

- ▶ **Name** - The type of PDFmark.

```
/ANN pdfmark
```

The full string passed to the `AddPDFMark` method appear as:

```
[ /Rect [ 75 586 456 663 ] /Contents (Note Type Comment Example.) /ANN pdfmark"
```

This creates a PDF Comment in the output PDF at the designated location.

Development Examples

You can use activePDF Server in any COM enabled environment. The following examples demonstrate how Server is implemented with several common applications and languages.

- **Word - VBScript**
- **Word - ASP**
- **Visual FoxPro**
- **Clarion**
- **Delphi**

Word - VBScript

The following VBScript example demonstrates how to convert a Microsoft® Word document to PDF.

```
' Get local path as varPath
arrayScr = Split(WScript.ScriptFullName, "\", -1, 1)
For i = 0 to UBound(arrayScr) - 1
    varPath = varPath & arrayScr(i) & "\"
Next

' Define Variables
' PDF output name
varPDFName = "word.pdf"

' Word File To Convert
varWordFile = "word.doc"

' Timeout for Wait
varTimeout = 60

' Instantiate Object
Set APS = CreateObject("APServer.Object")

' Set Server Properties
APS.OutputDirectory = varPath
APS.NewDocumentName = varPDFName

' Tell Server to begin the print process
varReturn = APS.StartPrinting()
If varReturn <> 0 Then Error("StartPrinting") End If

' Word Automation
Set WordObject = CreateObject("Word.Application")
WordObject.DisplayAlerts = False
Set NewDoc = WordObject.Documents.Open((varPath & varWordFile), False, True)
Set WordDialog = WordObject.Dialogs(97)
WordDialog.Printer = APS.NewPrinterName
WordDialog.DoNotSetAsSysDefault = 1
WordDialog.Execute
NewDoc.PrintOut False
NewDoc.Close False
WordObject.Quit False
Set WordObject = Nothing

' Tell Server to end the print process
varReturn = APS.StopPrinting()
If varReturn <> 0 Then Error("StopPrinting") End If

' Wait for the conversion results
varReturn = APS.Wait(varTimeout)
If varReturn <> 0 Then Error("Wait") End If

' Clear Server Object
Set APS = Nothing

' Conversion Successful
Msgbox "Done!"

' Error Handling
Sub Error(Method)
    MsgBox "" & Method & "" failed with a "" & varReturn & _
```

```
"" & VBCRLF & "KB article on Return codes:" & VBCRLF & _  
"http://www.activepdf.com/support/knowledgebase/viewKb.cfm?id=10549&tk=ts"  
Set APS = Nothing  
Wscript.Quit  
End Sub
```

Word - ASP

The following ASP example demonstrates how to convert a Microsoft® Word document to PDF.

```

<%
' Define Variables
' Local Path
varPath = Server.MapPath(".")

' PDF output name
varPDFName = "word.pdf"

' Word File To Convert
varWordFile = "word.doc"

' Timeout for Wait
varTimeout = 60

' Instantiate Object
Set APS = Server.CreateObject("APServer.Object")

' Set Server Properties
APS.OutputDirectory = varPath
APS.NewDocumentName = varPDFName

' Tell Server to begin the print process
varReturn = APS.StartPrinting()
If varReturn <> 0 Then Error("StartPrinting") End If

' Word Automation
Set WordObject = Server.CreateObject("Word.Application")
WordObject.DisplayAlerts = False
Set NewDoc = WordObject.Documents.Open((varPath & "\" & varWordFile), False, True)
Set WordDialog = WordObject.Dialogs(97)
WordDialog.Printer = APS.NewPrinterName
WordDialog.DoNotSetAsSysDefault = 1
WordDialog.Execute
NewDoc.PrintOut False
NewDoc.Close False
WordObject.Quit False
Set WordObject = Nothing

' Tell Server to end the print process
varReturn = APS.StopPrinting()
If varReturn <> 0 Then Error("StopPrinting") End If

' Wait for the conversion results
varReturn = APS.Wait(varTimeout)
If varReturn <> 0 Then Error("Wait") End If

' Clear Server Object
Set APS = Nothing

' Conversion Successful
Response.Write "Done!"

' Error Handling
Sub Error(Method)
    Response.Write "' ' & Method & "' failed with a "' & varReturn & "' <br>"

```

```
Response.Write "<a href='http://www.activepdf.com/"
Response.Write "support/knowledgebase/viewKb.cfm?id=10549&tk=ts'"
Response.Write "KB Article for Return Codes</a>"
Set APS = Nothing
Response.End
End Sub
%>
```

Visual FoxPro

The following Visual FoxPro® example demonstrates how to convert a report to PDF.

```
close databases
use myfile
X = CREATEOBJECT("APSERVER.OBJECT")
Z = X.STARTPRINTING()
set printer to X.NewPrinterName
report form myReport to printer
X.StopPrinting
I = X.Wait(30)
```

Clarion

The following steps demonstrate how to convert use activePDF Server with Clarion®.

1. Create a **Window**.
2. Add the following **Control**:

```
Control OLE,AT(100,100,100,100),Use(?APServer)
```

3. Add the following code:

```
?APServer{Prop:Create} = "APServer.Object"  
z# = ?APServer{"StartPrinting"}
```

4. Create a **Report**.
5. Add the following code::

```
Printer{PROP:Device} = ?APServer{"NewPrinterName"}
```

6. Print your **Report**.
7. Add the following code:

```
I# = ?APServer{"Wait(30)"}
```

Delphi

The following steps demonstrate how to convert use activePDF Server in Delphi®.

1. Open your development tool, and create a new project.
2. On the **Project**, click **Import Type Library**.
3. Select the **activePDF Server OLE Control module**, and then click **OK**.
4. Add a **new unit** to the **uses** clause.
5. For the **procedure** or **function**, add a variable for the **Type**. For example, **APS**.
6. Add the **StartPrinting** method to your code.

```
r := APS.StartPrinting;
```

7. Do one of the following:
 - ▶ Declare a **TPrinter** variable.
 - ▶ Use the built-in **Printer** object.
8. Add the remainder of the Server code.

```
Printer.PrinterIndex = Printer.Printers.IndexOf(APS.NewPrinterName);  
Printer.BeginDoc;  
// Do your printing thing  
APS.StopPrinting;  
r := APS.Wait(30);
```

Server Updates

The following updates represent those resulting from the submission of an activePDF Technical Support Ticket.

- ▶ 3.8.3
- ▶ 3.8.2
- ▶ 3.8
- ▶ 3.5.2 SP7
- ▶ 3.5.2 SP6
- ▶ 3.5.2 SP5A
- ▶ 3.5.2 SP5
- ▶ 3.5.2 SP4A
- ▶ 3.5.2 SP4
- ▶ 3.5.2 SP3
- ▶ 3.5.2 SP2
- ▶ 3.5.2 SP1

NOTE: If you received an incident number, but do not see a corresponding update, please submit a new ticket to [activePDF Technical Support](#) including the previous ticket or incident number.

3.8.3

- ▶ FIX: Issue where Type1 Base14 Fonts not being embedded correctly.
- ▶ FIX: Issue where activePDF registry key incorrectly set as 'ActivePDF'.
- ▶ FIX: Issue where Color and Gray compression settings were not being properly set during conversion.
- ▶ FIX: Issue where persistent printer ports not being properly reused and removed.

3.8.2

- ▶ Resolved an issue, whereby WebGrabber 3.5.2 SP7 would not function with Server 3.8.
- ▶ Resolved an issue, whereby using Toolkit 4.0.4 or later on an evaluation Server pdf, resulted in corrupted output.
- ▶ Resolved an issue where DocConverter and Webgrabber 3.5.2 were not correctly passing conversion parameters to Server 3.8.
- ▶ Resolved an issue where conversion engine unable to access files referenced in the file to be converted.
- ▶ Latest version (1.1.165) of SecureLM License Administrator included in Server installation. Selecting Uninstall License from the Activate menu will now auto-deactivate the license as well.

3.8

- ▶ Complete architectural restructure that significantly reduces the number of processing steps required while increasing reliability and consistency.
- ▶ A new, easy-to-use interface for the configuration editor.
- ▶ The ability to save and load specific server configurations.
- ▶ Improved logging capabilities to ensure optimal document fidelity.
- ▶ Enhanced fault tolerance with administration rights required for installation.
- ▶ Legacy support for all existing methods and complete documentation of all new methods is available in the updated user manual.
- ▶ The activePDF License Administrator has been introduced to facilitate license management and validation.
- ▶ Static and Reserved printers have been discontinued.
- ▶ The following property has been deprecated: *GeneralFlags*
- ▶ The following ImageToPDF parameters have been deprecated: *Width*, *Height*, *PersistRatio*, *PaperSize*, *OffsetX*, *OffsetY*
- ▶ The following methods have been added: *SetColorDownsampleThreshold*, *SetGrayDownsampleThreshold*, *SetMonoDownsampleThreshold*

- ▶ The following properties have been added: *ASCIIEncode*, *AutoRotate*, *ColorDownsampleType*, *ColorFilter*, *CompatibilityLevel*, *EmbedAllFonts*, *FlateCompression*, *FontSubsetting*, *GrayDownsampleType*, *GrayFilter*, *MonoDownsampleType*, *MonoFilter*, *PreserveHalftone*, *PreserveOverprint*, *UCRandBGR*
- ▶ Fixed *FingerprintPDF* and *IsFingerprintValid* methods.
- ▶ Fixed extra page added with PostScript® (PS) conversion.
- ▶ Fixed Scrunched fonts in output from PS to PDF conversion.
- ▶ Fixed *ImageToPDF* failing to size TIFF to PaperSize or pages appearing as negative when Toolkit is installed on the machine.
- ▶ Fixed *ImageToPDF* converting only the first page with certain TIFF images.
- ▶ Fixed *ImageToPDF* failing to use the correct page size with certain TIFF images.
- ▶ Fixed APSENET.DLL to show correct version number.
- ▶ Fixed issue with *LoadRemoteProfile* returning a -1.
- ▶ Fixed Euro symbol embedding to correctly display and is searchable.
- ▶ Fixed the use of Custom fonts with *PSToPDF*.

3.5.2 SP7

- ▶ Added new optional function in the COM interface called *GetExclusiveAccess* to address a latency issue on multi-processor machines, occurring between the completion of the creation of a PostScript® file and the execution of WSTPDF.EXE.
- ▶ Added sever options to the *Configuration Editor* to permit updating of most activePDF Server registry entries.
- ▶ Added new *EnableWatcher* registry entry that replaces the *IgnoreWatcher* flag, which is now turned off by default and should not be turned on unless explicitly advised by activePDF Technical Support.
- ▶ Added an option for file information to be read in memory rather than written to disk in the form of APX files.
- ▶ Added a registry setting called *VMSize* to enable increasing the amount of virtual memory allocated during conversion.
- ▶ Fixed an issue with MFC class libraries and hyper-threading, specifically deleting thread local data when the application closes normally, causing an application popup window to occur as a result of a *Memory could not be read* error occurring in WSPTPDF.exe.
- ▶ Fixed an issue with APSENET.exe leaking handles after every document processed.
- ▶ Fixed an issue with thread count for Microsoft Windows® Server 2003 where CPU utilization incrementally increased due to stacked APSENET and isass.exe processes.

3.5.2 SP6

- ▶ Added support for the Windows Server 2003 operating system.
- ▶ Fixed a document specific issue where lines and text found in PostScript® file are missing from the output PDF. Caused by a defect in the PDF Library.
- ▶ Fixed a document specific issue where the reset command in the PostScript is not accepted during conversion. Caused by a defect in the PDF Library.
- ▶ Fixed a document specific issue where kerning in the PostScript is missing from the output PDF. Caused by a defect in the PDF Library.

3.5.2 SP5A

- ▶ Added a registry value to enable or disable font substitution and embedding of base14 fonts.
- ▶ Added support for 128-bit security, including the *SetOutputSecurity128* method.

3.5.2 SP5

- ▶ Fixed an issue where the use of the *AddFileBookmark* method creates extra symbols in the PDFMark stream, preventing the file from processing.
- ▶ Fixed a document specific issue where PostScript files lacking the %EOF fail to convert. Caused by a defect in the PDF Library.