AcroPlot Auto 2024

AcroPlot Auto takes conversion one step further and allows you to automatically convert entire drives to PDF or DWF on a continuous basis or at a scheduled time each day. Create watched project folders for AcroPlot Auto to monitor at various intervals and create updated PDF, DWF, or TIFF files daily.

AcroPlot Auto does provide for the same high quality image output as AcroPlot Repro so many of our Reprographics customers have found it useful to use AcroPlot Auto to watch folders and convert the files automatically and then load the image files into their Reprographics program.

2024
Last Updated: September, 21, 2023

Table of Contents

Our Products

Thank you for using the AcroPlot family of software products. We believe that CADzation software solutions are some of the most powerful applications for converting engineering documents into PDF, DWF, or Image files. Since 2001, CADzation has been providing thousands of clients the highest quality conversion solutions that also result in the smallest file sizes for PDF and DWF. CADzation was the first company to support "lines merge" functionality when converting from AutoCAD DWG files to PDF, and is also the first company to provide conversion to both PDF and DWF from the same software application.

However, we are never resting on our past accomplishments and are constantly looking for ways to improve our programs. We can best accomplish this with your help. So, please feel free to comment on our software or provide any suggestions for new features or improvements. The best way to do this is through our online support desk at http://www.cadzation.com/support.htm.

AcroPlot Suite (Includes AcroPlot Pro, AcroPlot Matrix, and High Quality PDF Printer Driver)

AcroPlot Suite is has been one of the most advanced CAD to PDF solution available on the market since 2001. With AcroPlot Pro to convert various formats to PDF, DWF, and image formats, AcroPlot Matrix to edit and markup PDF files, and one of the highest quality PDF Printer Drivers we have all of your needs covered.

AcroPlot Pro is the next generation in digital CAD conversion software. With the capability to convert an extensive list of file formats to PDF, DWF, and images AcroPlot Pro will serve as a nexus for CAD Professionals everywhere.

Input Formats: DWG, DXF, DOC, TXT, RTF, XLS, BMP, CALS, JPG, GIF, TIFF, PLT and more.

Output Formats: PDF, DWF, BMP, CALS, JPG, GIF, GP4, PNG, TIFF
More Information Online-->

AcroPlot Matrix software enables you to easily view, comment, modify and print an extensive list of file types like PDF, DWF, TIFF, PLT, and more, that are associated with the architecture, engineering, construction, and manufacturing design workflows in today's business environment.

With a streamlined interface that makes it easy to use, yet very powerful, AcroPlot Matrix doesn't get in the way of getting work done. Users can quickly open and add comments or stamps to files, insert, reorder, rotate, or delete pages, and even merge various file types into a single PDF or DWF.

Printing files to scale at full or reduced sizes that are dimensionally accurate is now possible in AcroPlot Matrix by using the "Custom Scale" option in the printing dialogue window. While other PDF and DWF software applications only print to

either "No Scale" or "Reduce to Printable Area", AcroPlot Matrix provides the functionality to define a true reduction percentage so that output to any printer at any scale is now possible -- half-size prints are now quick, easy and accurate.

AcroPlot Matrix is quickly becoming the preferred software application for PDF communication across entire organizations because it is so flexible and easy to use, as well as, cost effective to deploy. It works well in all departments like administrative, estimating, accounting, legal, parts, maintenance, etc.!
More Information Online-->

AcroPlot Auto takes conversion one step further and allows you to automatically convert entire drives to PDF or DWF on a continuous basis or at a scheduled time each day. Create watched project folders for AcroPlot Auto to monitor at various intervals and create updated PDF, DWF, or TIFF files daily.
More Information Online-->

AcroPlot Auto does provide for the same high quality image output as AcroPlot Repro so many of our Reprographics customers have found it useful to use AcroPlot Auto to watch folders and convert the files automatically and then load the image files into their Reprographics program.

AcroPlot Repro - CADzation's AcroPlot line of software has always been able to produce some of the highest quality PDF files on the market. But when enabling the "lines merge" in PDF files the software that does the final rendering (Ghostscript or Adobe) to create the output for the plotters have had a hard time converting the large format files because they are not optimized for large format drawings.

AcroPlot Repro solves this problem. Faster processing, better dithering, and the ability to actually convert files at higher dpi (which currently Ghostscript seldom does) means you can now create the same high quality output on your plotter as you see in the PDF file.
More Information Online-->

Since Several of our Products are included in each other here is a table of the basic features.

Features	AcroPlot Suite	AcroPlot Auto	AcroPlot Repro
Batch convert AutoCAD to PDF	X	X	
Batch convert Word to PDF	X	X	
Batch convert AutoCAD to PDF, DWF, and Images	X	X	

Batch convert Excel to PDF, DWF, and Images	X	X	
Batch convert PDF to DWF or Images	X	X	
Batch convert DWF to PDF and Images	X	X	
Batch convert Images to PDF and DWF	X	X	
Batch convert PLT files to PDF, DWF, and Images	X	X	
Includes the PDF-XChange for AcroPlot Pro Printer Driver to print to PDF from any Windows Appliaction.	X	X	
Includes AcroPlot Matrix for markup and editing of PDF files as well as converting DWF, PLT, and Image files to PDF.	X	X	
Converts PLT, PDF, and DWF files to High Quality TIFF files for plotting.		X	X
Watches folders and converts entire drives of supported files to PDF, DWF, and Images.		X	

Style Conventions

The topics in this Help system are designed to give you concise overviews of the software's functions and step-by-step procedural information. In the course of this instruction you'll encounter a number of conventions:

• Words in **bold** text refer to parts of the user interface. For example, "Click **Rotate Page**."

• You'll notice that some words are <u>underlined and highlighted in blue</u>. Clicking on one of these links will take you to a screen with additional information.

• **"Choose"** and the **">"** symbol are used to refer to opening menus. For example, "Choose **File > Save**" means open the **File** menu and select the **Save** icon.

• "Press **Alt+Ctrl**" means press and hold the Alt key as you press the Control key.

• To "hover over" something means to move the cursor over it and linger there to get additional information.

• To "Drag" something means to move cursor over it, hold down the left mouse button, and drag the mouse. Release the mouse button to stop.

In addition many of the screen shots are annotated with special symbols that, when hovered over, displays a window with further details on the subject matter. Try it below:

Example

Hover Over This

Displays some more detail, the exact location of the topic in question and sometimes more pictures.

Another Detailed Topic

With another picture to help explain this topic

System Requirements

• Fully supported on 32 and 64 bit versions of Windows 7 through Windows 11.

• Fully supported on 32 and 64 bit versions of Windows Server 2012 through Windows Server 2023. Windows Server requires the Windows Server with Desktop Experience feature to be installed.

• You should also have the latest Windows updates and service packs installed for your operating system.

• 500 MB of free disk space on the C: drive.

• 2 GB of RAM minimum. (4 GB or higher strongly recommended.)

• Pentium 1 GHz or better processor minimum. (Pentium 1.8 GHz or higher recommended.)

• 1024 x 768 with true colors video display minimum.

• Works without AutoCAD or with AutoCAD 2000-2024

AcroPlot Auto Overview

Installing AcroPlot Auto

To install AcroPlot Auto first run the acroplotsetup.exe (it may also contain the release date like acroplotsetup-2013-01-16.exe) in the AcroPlotAuto.zip file. This will install the AcroPlot Pro software first.   Then copy all of the remaining files in the AcroPlotAuto.zip file over to the install directory of the AcroPlot Pro software that contains the AcroPlot.exe file.

On 64 bit Windows this is typically:
C:\Program Files(x86)\AcroPlot
!!!Note that on 64 bit Windows AcroPlot installs the 64 bit Printer driver into C:\Program Files \AcroPlot but you will not find the AcroPlot.exe file there. Do not copy the files there, please copy them to the location of the AcroPlot.exe file.

On 32 bit Windows this is typically
C:\Program Files\AcroPlot

Getting Started

Since AcroPlot Auto is not really designed as an end user program please contact rmcmanamy@cadzation.com with any questions regarding the use of AcroPlot Auto. We will highlight some of the basic settings here but please just contact us if you have any questions or suggestions.

Using Startup Switches

AcroPlot Auto accepts the following startup switches which will allow you to automatically start AcroPlot Auto. You can use the Windows Scheduling to start it at a specific time or use the Windows Startup to start it whenever you reboot the computer.

/a = Automatically start the conversion

/m = Minimize the dialog box.

So you would call it like AcroplotAuto.exe /a /m

Creating Scan Settings

Basically you can define as many different scan settings as you would like and use different conversion settings for them. The conversion settings are first defined by using AcroPlot Pro and saving the settings as a Named Setting. To create a new Scan Settings simply hit the Blue "+" button. You can also turn on and off the various Scans that you have defined as well as set the time intervals at which you would like the conversion to take place. The way that the time intervals work is that we start scanning at the top of the list the first time through and then after each scan the next time that the scan should take place is recorded.

When are Files Converted or Reconverted

If the output file(pdf) is not located in the output directory then the input file(dwg) will be converted. Or if the last modified date of the input file(dwg) is newer than the created date of the output file(pdf) then it will also be converted.

Reconverting all files

If you need to reconvert all files for any reason you can select the Reconvert All Files from the Setup menu.

Converting with our Viewer Instead of AutoCAD.

To convert with our viewer instead of AutoCAD you must select the Use Internal Viewer to Convert DWG in the Setup menu.

Source Directories and Files to Convert

You can only select one source drive or directory per scan setting. To do a different drive or directory you just create another scan setting. The Source Drive or Directory must be an absolute path and cannot contain wildcard characters. The Files to Convert on the other hand can contain wildcard characters and you can use directories in the files to covert to make sure that you only convert certain subdirectories. Say all of your files are in C:\Projects but you only want to convert the drawing files that reside in the \Release subfolders for each project. You could set the Source Drive or Directory to C:\Projects and then the Files to Convert to *\Release*.dwg. This would then convert files located in subdirectories like:

C:\Projects\C1000\Release\C1000-10-01.dwg

C:\Projects\C1000\Release\30Percent\C1000-10-01.dwg

C:\Projects\Release\C1000-10-01.dwg

Output Folder

AcroPlot Auto supports several fields that you can use to build the proper output folders based on the source filename that is being converted.

A few of the most common cases are.

1. **You want to output file placed in the same folder as the source file.**
   For this option just leave the Output Drive or Folder textbox blank.

2. **You want to locate the file to a folder created in the same folder as the source file.**
   For this option just add in a backslash and the folder name into the Output Drive or Folder textbox. This works well if you want to create PDF, DWF, and Image files of the source file.
   Example: \PDF
   Example: \DWF
   Example: \TIFF

3. **You want to output the file to a different drive with the same folder structure.**
   For this option just add in the drive letter followed by \||DIR|| into the Output Drive or Directory textbox. We will then replace the ||DIR|| field with the folder of the source file.
   Example: L:\||DIR||

4. **You want to locate the file to a different drive with the same folder structure but also with a sub folder added.**
   For this option you would combine examples 3 and 2.
   Example: L:\||DIR||\PDF
   Example: L:\||DIR||\DWF
   Example: L:\||DIR||\TIFF

More complex output directories can be created by using the following fields in the Output Drive or Directory value. These also work in the Thumbnail Drive or Directory value.

	DRIVE	
	DATE	
	DIR	
	DIR1	
	DIR2	
	DIRMINUS1	
	DIRMINUS2	
	SETTINGNAME	
	RELATIVE	
	RELATIVE1	

||RELATIVE2||
||RELATIVEMINUS1||

The relative fields are based on the relative folder structure from the Source Drive or Directory value. In other words we strip off the Source Drive or Directory from the actual location where it was found and calculate the values from the remaining.

Now given that the source file was found in say:
C:\Projects\Construction\C1000\C1000-10\C1000-10-01.dwg
The Fields would equal:
||DRIVE|| - "C:\"
||DATE|| - "2005-01-30"
||DIR|| - "Projects\Construction\C1000\C1000-10"
||DIR1|| - "C1000-10"
||DIR2|| - "C1000"
||DIRMINUS1|| - "Projects\Construction\C1000"
||DIRMINUS2|| - "Projects\Construction"

Now depending on what you had used for the Source Drive and Directory the Relative values will change.
If you used "C:\" for the Source Drive or Directory and the file was found as C:\Projects\Construction\C1000\C1000-10\C1000-10-01.dwg
The Fields would equal:
||RELATIVE|| - "Projects\Construction\C1000\C1000-10"
||RELATIVE1|| - "C1000-10"
||RELATIVE2|| - "C1000"
||RELATIVEMINUS1|| - "Projects\Construction\C1000"
||RELATIVEMINUS2|| - "Projects\Construction"

If you used "C:\Projects" for the Source Drive or Directory and the file was found as C:\Projects\Construction\C1000\C1000-10\C1000-10-01.dwg
||RELATIVE|| - "Construction\C1000\C1000-10"
||RELATIVE1|| - "C1000-10"
||RELATIVE2|| - "C1000"
||RELATIVEMINUS1|| - "Construction\C1000"
||RELATIVEMINUS2|| - "Construction "

If you used "C:\Projects\Construction" for the Source Drive or Directory and the file was found as C:\Projects\Construction\C1000\C1000-10\C1000-10-01.dwg
||RELATIVE|| - "C1000\C1000-10"
||RELATIVE1|| - "C1000-10"
||RELATIVE2|| - "C1000"
||RELATIVEMINUS1|| - "C1000"
||RELATIVEMINUS2|| - ""

Deleting Original Files

To unlock the ability to delete the original files type in the code below in the Setup->Options menus.

DELETEUNLOCKCODE-1773430238

Please note this does just what it says and it DELETES the original file after the conversion.

Running AcroPlot Auto as a Service

Many programs like AutoCAD and Microsoft Office that we are using to perform the conversions are not designed to run as a service and hence they need to be logged in as a specific user to run and typically need interaction with the desktop also.

We have found that you can use a small program called FireDaemon to get these programs working with AcroPlot Auto as a service. The program is only $49 as of this writing and well worth the price. Click here to visit the www.firedaemon.com site for more information.

1.) To setup AcroPlot Auto to run as a service in FireDaemon after starting FireDaemon simply add a new service.

2.) Next you can set the Short Name and Display Names to AcroPlot Auto

3.) Browse to the Executable AcroPlotAuto.exe typically in the C:\Program Files\AcroPlot or C:\Program Files (x86)\AcroPlot Folder

4.) VERY IMPORTANT!! You must add the /a command line parameters into the Parameters section as highlighted below. This is the command line parameter in AcroPlot Auto to tell it to automatically start converting your saved jobs.
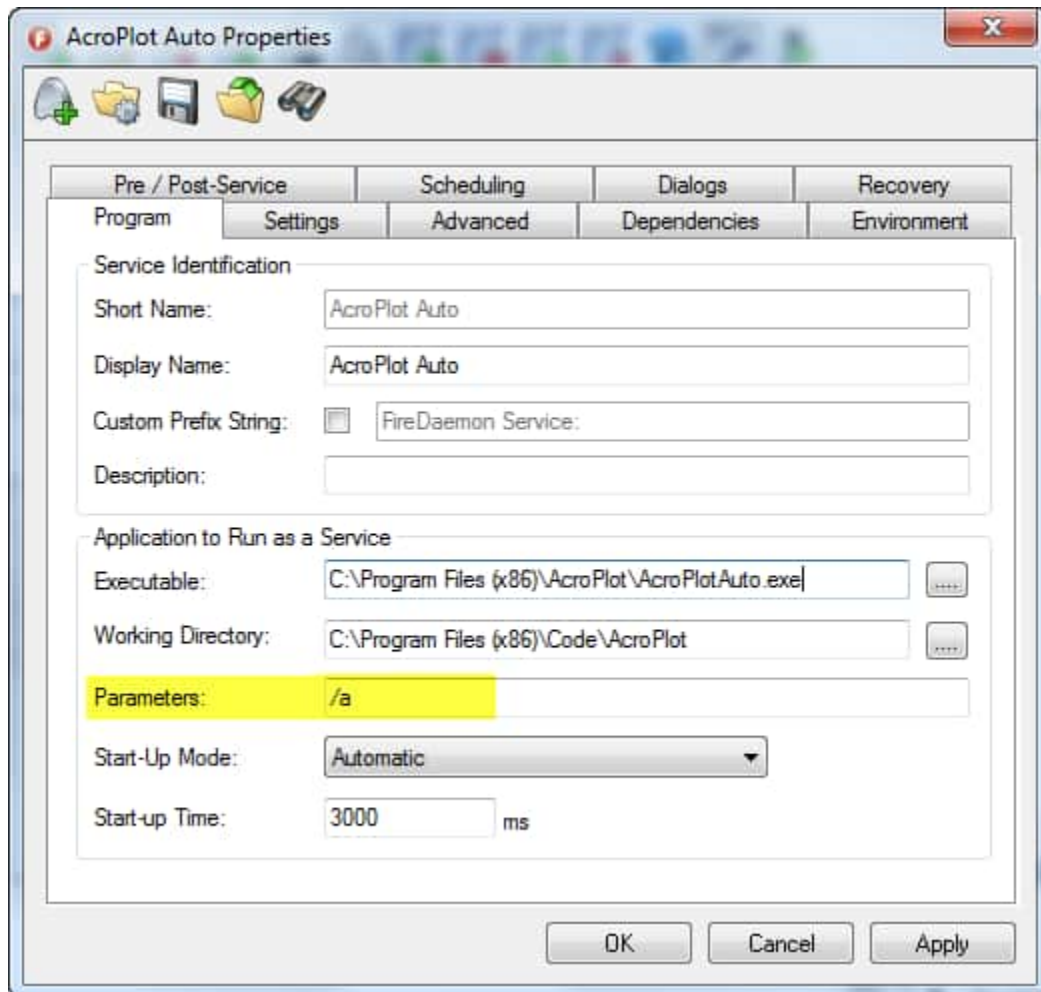
5.) Now switch to the Settings Tab. From here you want to specify the Logon Account and passwords to use. Remember, AutoCAD and Microsoft Office will not know your user preferences and will not run without specifying a user to run.

6.) VERY IMPORTANT!! You must also check the Interact with Desktop checkbox as highlighted below.
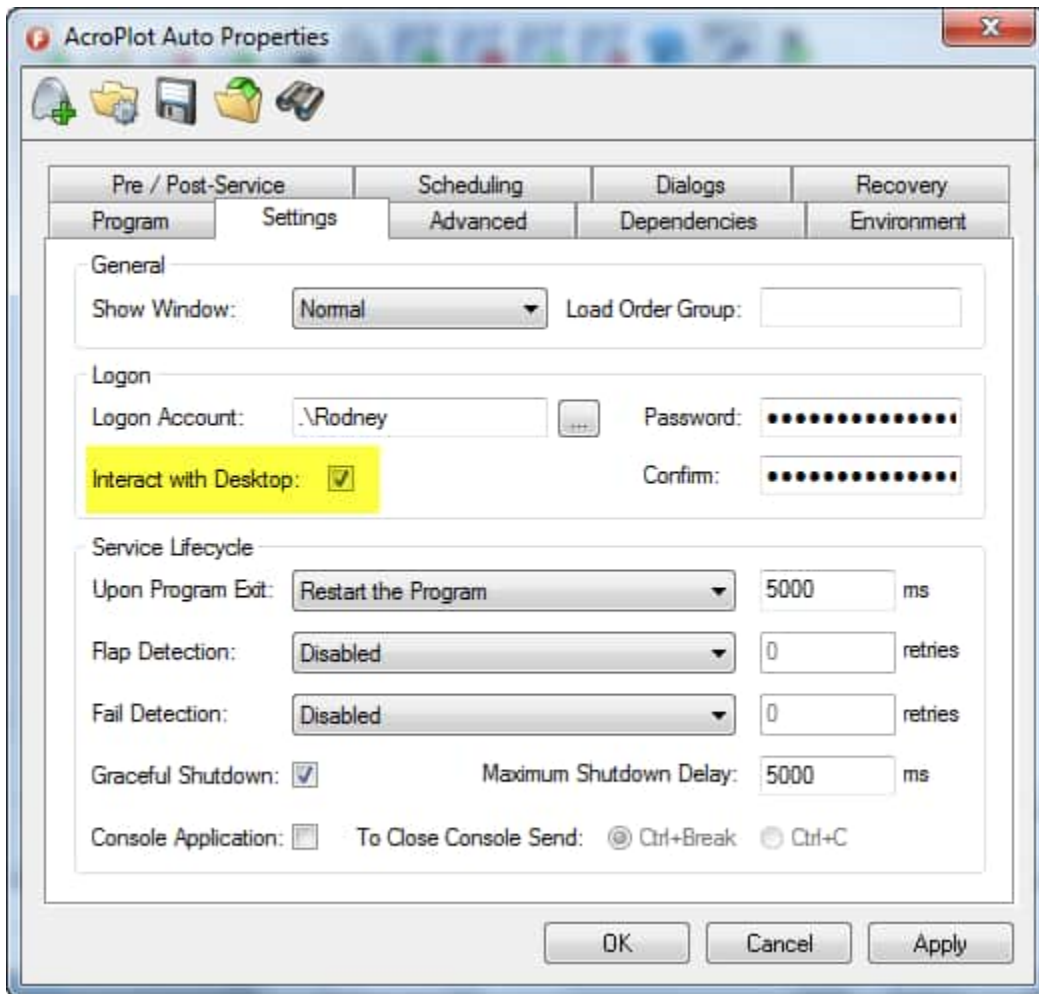
7.) All other default settings can stay the same.

8.) You can also use the Scheduling Tab of FireDaemon if you want to schedule it to start at a specific time. Please see the FireDaemon help for more information on their features and settings.

9.) VERY IMPORTANT!! Since FireDaemon is basically tricking AcroPlot Auto, AutoCAD, and Microsoft Office to run as a service and since AutoCAD and Microsoft Office need to interact with the Desktop it may cause the Microsoft Interactive Detection Service to display dialogs which may halt processing. Because of this you will probably want to disable the Interactive Detection Service in the normal Windows Services dialog.

Setting AutoCAD to Interact with The Desktop

AutoCAD was not designed to run as a service. Although our AcroPlot Auto will run as a service there may be cases where in order to get AutoCAD to function properly without using the Firedaemon software we need to set it to run as the Interactive user.

From the Windows Start Menu do Start->Run Then type in DCOMCNFG and hit enter.   This Should Start the component services.
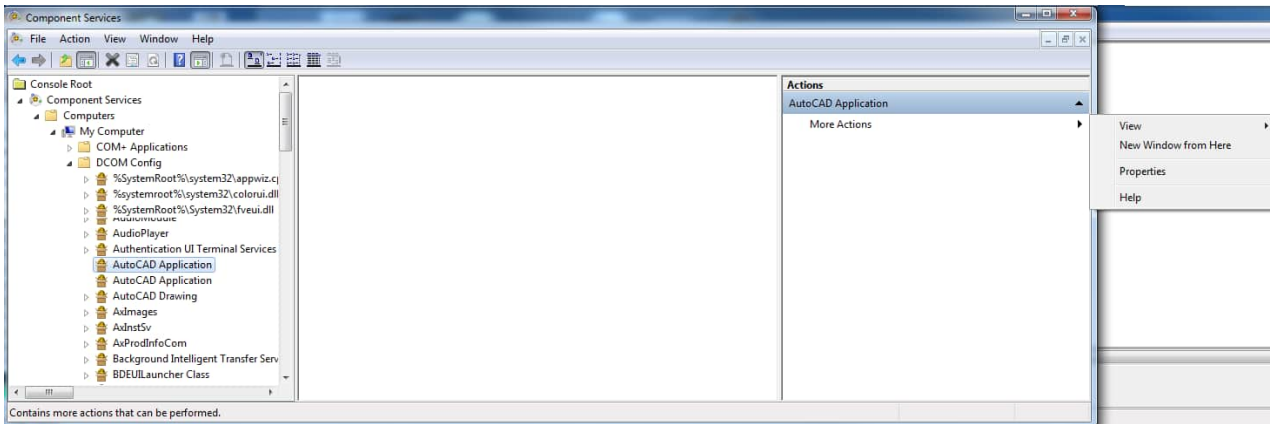
ActiveX and Command Line

Overview

With your AcroPlot Auto license you can also user our AcroPlot ActiveX (COM) interface or the Command Line calls to the AcroPlot.exe to perform your conversions.

This is popular for users who would like to integrate our conversions with their existing document management systems.

AcroPlot Auto cannot be combined into another software package for resale without a signed agreement by CADzation and without purchasing additional licenses for each server that AcroPlot Auto is installed on.

AcroPlot Standard API

The Standard API designed for users who would like to create their own toolbars or menus to access AcroPlot's basic functionality with predefined output filenames.

All AcroPlot Suite licenses can access some of the Standard AcroPlot API from inside of AutoCAD to provide predefined output for where they would like the files created to be saved.

To create or convert files using the command line mode API or ActiveX automation mode interface, it is required to purchase a license of AcroPlot Auto for each server running the process. If there will be additional PC's running AcroPlot Suite licenses or any of its components, then it is also required to purchase the number licenses needed in addition to the AcroPlot Auto server license.

Notes:

1. If the sOutputFilename exists the PDF files will be merged to it.
2. If the sOutputFilename is read-only or is opened in Adobe Acrobat or Adobe Reader the conversion will fail.
3. If the RestoreSettings is not called or provided a proper settings name then the last used settings in AcroPlot Jr. will be used.
4. If you are merging multiple layouts or drawings into the same PDF make sure that you turn off the OpenPDF setting by using:

SetSetting "AcroPlotSettings.OpenPDF" 0

The Standard API Calls are:

AcroPlotJCurrent(ByVal *oAcadApp* As AcadApplication) As Boolean
This method will convert the current layout (either modelspace or paperspace) to PDF.

AcroPlotJModel(ByVal *oAcadApp* As AcadApplication) As Boolean
This method will convert the modelspace layout to PDF.

AcroPlotJPaper(ByVal *oAcadApp* As AcadApplication) As Boolean
This method will convert the all of the paperspace layouts in the tab order to PDF.

AcroPlotJView(ByVal *oAcadApp* As AcadApplication) As Boolean
This method will convert the modelspace layout to PDF.

GetAvailableSettings(ByVal sSettingName As String, ByRef sRet As String) As Long
This method will return a string array of the available setting names based on the following 3 categories that can be passed in for sSettingCategory
AcroPlotSettings
AcroPlotPrinter
AcroPlotAutoCAD

GetSetting(ByVal sSettingName As string, ByRef vValue As Variant) As Long
This method allows you to get the current value of the setting name that you passed in. If the setting name does not exist it will return 0. If it does exist it will return 1.

Init() As Long
This method initializes the AcroPlot API.

Quit()
This method must be called to release the AcroPlot Application Object properly.

RestoreSettings(ByVal *sSettings* As String) As Long
This method restores a named setting that you had previously saved using AcroPlot or AcroPlot Jr.

RunAutoCADFull(ByVal oAcadApp As Object) As Boolean
Runs the AcroPlot Program from inside of AutoCAD. Useful if you want to provide your own menu routines.

RunAutoCADJunior(ByVal oAcadApp As Object) As Boolean
Runs the AcroPlot Junior Program from inside of AutoCAD. Useful if you want to provide your own menu routines.

RunAutoCADPro(ByVal oAcadApp As Object) As Boolean
Runs the AcroPlot Pro Program from inside of AutoCAD. Useful if you want to provide your own menu routines.

SetOutputFilename(ByVal *sOutputFilename* As String) As Long
This method allows you to set the output filename. **The user will still be prompted for the filename unless you are an AcroPlot Auto customer or an AcroPlot Pro customer with 25 or more licenses.**

SetOutputFormtID(ByVal lOutputFormat As Long) As Long
This method allows you to set the output format to use. While the default output format is automatically set when you call the SetOutputFIlename there are some formats like Tiff that can be color or monochrome as well as different output DPI. **So you must call the SetOutputFormatID AFTER the SetOutputFilename.**
The valid output formats values are:
  pdf = 1
  dwf6 = 2
 'reserved for future use
  BMP = 4
  CAL = 5
  GIF = 6
  gp4 = 7
  JPG = 8
  PNG = 9
  tif = 10 (color tiff at 100 dpi)
  tifmono = 11 (monochrome group 3 fax format tiff at 100 dpi)
  tifmono200 = 12 (monochrome group 3 fax format tiff at 200 dpi)
  tifmono300 = 13 (monochrome group 3 fax format tiff at 300 dpi)
  tifmono400 = 14 (monochrome group 3 fax format tiff at 400 dpi)
  tif200 = 15 (color tiff at 200 dpi)
  tif300 = 16   'not supported reserved for future use
  tif400 = 17   'not supported reserved for future use

dwfpdf = 18
dwfpdftif = 19
dwfpdftifmono = 20
dwfpdftifmono200 = 21
dwfpdftifmono300 = 22
dwfpdftifmono400 = 23
pdftif = 24
pdftifmono = 25
pdftifmono200 = 26
pdftifmono300 = 27
pdftifmono400 = 28
'reserved for future use
rasterpdf100 = 30   'not supported reserved for future use
rasterpdf150 = 31   'not supported reserved for future use
rasterpdf200 = 32   'not supported reserved for future use
rasterpdf300 = 33   'not supported reserved for future use
rasterpdf400 = 34   'not supported reserved for future use
rasterpdfmono100 = 35   'not supported reserved for future use
rasterpdfmono150 = 36   'not supported reserved for future use
rasterpdfmono200 = 37   'not supported reserved for future use
rasterpdfmono300 = 38   'not supported reserved for future use
rasterpdfmono400 = 39   'not supported reserved for future use
tif600 = 40   'not supported reserved for future use
tifmono600 = 41   (monochrome group 3 fax format tiff at 600 dpi) 'not supported reserved for future use
dwfpdftifmono600 = 42   'not supported reserved for future use
pdftifmono600 = 43   'not supported reserved for future use
rasterpdf600 = 44   'not supported reserved for future use
rasterpdfmono600 = 45   'not supported reserved for future use
cal200 = 46   'not supported reserved for future use
cal400 = 47   'not supported reserved for future use
cal600 = 48   'not supported reserved for future use
gp4200 = 49   'not supported reserved for future use
gp4400 = 50   'not supported reserved for future use
gp4600 = 51   'not supported reserved for future use
cal300 = 52   'not supported reserved for future use
gp4300 = 53   'not supported reserved for future use
dwfx = 54   'not supported reserved for future use

SetSetting(ByVal *sSettingName* As String, ByVal *vValue* As Variant) As Variant
This method allows you to override the settings from the RestoreSettings. See below for the currently supported settings.

Any setting in the AcroPlotSettings.ini file can be used used. You can typically find the file in the C:\Users \[USERNAME]\AppData\Roaming\CADzation\AcroPlot\AcroPlotsettings.ini location by using the name on the left as the SettingName like acroplotsettings_bookmarkacadblocknames.

Some common settings are:

sSettingName	Type	vValues

BackgroundPlot	Integer or Long	Controls if AutoCAD uses background plotting. On some systems like Server 2008 we sometimes need to enable background plotting to driver AutoCAD properly. Usually though you can leave this set to -1. -1 = Let us control background plotting needs 0 = Force off background plotting in AutoCAD 1 = Force on background plotting in AutoCAD
DWGCreateIndividualFiles	Integer or Long (True/False)	-1 = Create an individual file for each layout being converted 0 = Create a single file containing all layouts
AcroplotSettings.BookmarkAcadBlocknames	String	
AcroplotSettings.BookmarkAcadDrawingText	String	
AcroplotSettings.BookmarkAcadLayoutText	String	
AcroplotSettings.BookmarkDrawing	Integer or Long (True/False)	-1 = Add bookmarks into the PDF. 0 = Do not add Bookmarks into the PDF
AcroplotSettings.BookmarkDrawingIndividual	Integer or Long (True/False)	-1 = Add drawing level bookmark to the PDF. 0 =Do not add a drawing level bookmark to the PDF.
AcroplotSettings.BookmarkDrawingProperties	Integer or Long (True/False)	-1 = Add bookmarks for the drawing properties to the PDF 0 = Do not add bookmarks for the drawing properties to the PDF.
AcroplotSettings.BookmarkKeep	Integer or Long (True/False)	-1 = If an existing PDF file is being merged keep it's existing bookmarks instead of adding a new one. 0 = Remove all bookmarks and add in our calculated bookmark when merging PDF files.

AcroplotSettings.BookmarkLayout	Integer or Long (True/False)	-1 = Add a layout level bookmark into the PDF file. 0 = Do not add a layout level bookmark into the PDF file.
AcroplotSettings.BookmarkLength	Integer or Long	Also set the BookmarkUseLength to limit the number of characters in a bookmark, Maximum number of characters for the bookmark length.
AcroplotSettings.BookmarkMainText	String	
AcroplotSettings.BookmarkPDFText	String	
AcroplotSettings.BookmarkShowOnOpen	Integer or Long (True/False)	-1 = Show the bookmarks pane in the PDF when it is opened 0 = Do not force showing the bookmarks pane
AcroplotSettings.BookmarkUse	Integer or Long (True/False)	-1 = Add bookmarks to the PDF 0 = Do not add bookmarks
AcroplotSettings.BookmarkUseLength	Integer or Long (True/False)	Also set the BookmarkLength to limit the number of characters in a bookmark, -1 = Limit the length of the bookmark to the specified number of characters as defined by the BookmarkLength 0 = Do not limit the length of bookmarks.
AcroplotSettings.ConvertWith	String	This setting would only be used by the AcroPlot Advanced API users so please see that documentation for more information.
AcroplotSettings.CustomPlotScale	Double > 0.0	Also set the PlotScale = -101 setting when using the CustomPlotScale 1.0 = Full Size 0.5 = Half Size
AcroplotSettings.DwgOrientation	Integer or Long	1 = Portait 2 = Landscape 3 = Automatic
AcroplotSettings.DwgPaperSize	Integer or Long	1 = Letter 5 = Legal 3 = Tabloid 24 = C size sheet

		25 = D size sheet 26 = E size sheet 131 = Arch-A 132 = Arch-B 264 = Arch-C 265 = Arch-D 135 = Arch-E 266 = Arch-E1 360 = 22x33 259 = A0 260 = A1 66 = A2 8 = A3 9 = A4 11 = A5 70 = A6 12 = B4 13 = B5 361 = USER1 362 = USER2 363 = USER3 364 = USER4 365 = USER5 366 = USER6 367 = USER7 368 = USER8 369 = USER9 370 = Size saved in drawing 371 = Full Offset
AcroplotSettings.EnablePc3Support	Integer or Long (True/False)	-1 = Use pc3 files 0 = Do not use pc3 files
AcroplotSettings.DwfoutIncludeLayer	Integer or Long (True/False)	-1 = Include layer information 0 = Do not include layer information
AcroplotSettings.LinesMerge	Integer or Long (True/False)	-1 = Use lines merge 0 = Do not use lines merge Note: Not allowed with PDF/A-1a and PDF-A-1b formats
AcroplotSettings.OpenPDF	Integer or Long (True/False)	-1 = Open 0 = Do not open
AcroplotSettings.ReproBrightness	Integer or Long	-100 to 100 (Default = 0) Brightness setting to control making monochrome tiff files lighter or darker.

AcroplotSettings.ReproDitherType	Integer or Long	Used for converting files to monochrome tiff formats
AcroplotSettings.ReproScale	Double	Scale to use when converting to monochrome tiff files. >0.0 where 1.0 is full size and 0.5 is half size
AcroplotSettings.UseInternalViewerForPlotting	Integer or Long (True/False)	-1= Use the internal viewer instead of AutoCAD for plotting 0 = Use AutoCAD if installed for plotting
AcroplotAutoCAD.PlotScale	Integer or Long	0 = Scale to Fit 16 = 1:1 Full Size 100 = Saved -101 = Custom (Also set the AcroPlotSettings.CustomPlotScale)
AcroplotAutoCAD.PlotType	Integer or Long	1 = Extents 2 = Limits 3 = View 4 = Window 5 = Layout, -100 = Use saved Settings
AcroplotAutoCAD.PlotView	String	The view name to plot if PlotType = 3
AcroplotPrinter.Allow128Assembly	Integer or Long (True/False)	-1=When Allow128Bit is set to 2 this will allow users to assemble (add) PDF files to this one. 0=When Allow128Bit is set to 2 this will restrict users from doing it.
AcroplotPrinter.Allow128Bit	Integer or Long	0=Do not add password security 1=Add 40 bit password security 2=Add 128 bit password security Note: Not allowed with PDF/A-1a and PDF-A-1b formats
AcroplotPrinter.Allow128Extract	Integer or Long	-1=When Allow128Bit is set to 2 this will allow users to extract from the PDF file.

	(True/False)	0=When Allow128Bit is set to 2 this will restrict users from doing it.
AcroplotPrinter.Allow128InteractiveForms	Integer or Long (True/False)	-1=When Allow128Bit is set to 2 this will allow users to fill in interactive forms. 0=When Allow128Bit is set to 2 this will restrict users from doing it.
AcroplotPrinter.Allow128PrintLow	Integer or Long (True/False)	-1=When Allow128Bit is set to 2 this will allow users to only print low resolution output if the AllowPrint is also enabled. 0=When Allow128Bit is set to 2 this will restrict users from doing it.
AcroplotPrinter.AllowChanges	Integer or Long (True/False)	-1=When Allow128Bit is set to 1 or 2 this will allow users to make changes to the PDF. 0=When Allow128Bit is set to 1 or 2 this will restrict users from doing it
AcroplotPrinter.AllowCopy	Integer or Long (True/False)	-1=When Allow128Bit is set to 1 or 2 this will allow users to copy info from PDF. 0=When Allow128Bit is set to 1 or 2 this will restrict users from doing it
AcroPlotPrinter.AllowNotes	Integer or Long (True/False)	-1=When Allow128Bit is set to 1 or 2 this will allow users to add annotation to the PDF. 0=When Allow128Bit is set to 1 or 2 this will restrict users from doing it
AcroPlotPrinter.AllowPrint	Integer or Long (True/False)	-1=When Allow128Bit is set to 1 or 2 this will allow users to print the PDF. 0=When Allow128Bit is set to 1 or 2 this will restrict users from doing it
AcroPlotPrinter.Author	String	Specifies the author for the PDF Document Properties
AcroplotPrinter.GrayScale	Integer or Long	0 = Color 1 = Monochrome 2 = Grayscale
AcroPlotPrinter.Justification		
AcroPlotPrinter.Keywords	String	Specifies the keywords for the PDF Document Properties

AcroPlotPrinter.Orientation		
AcroPlotPrinter.PasswordAuthor	String	When Allow128Bit is set to 1 or 2 this will set the Author password to the text you entered. This is required when Allow128Bit is set to 1 or 2.
AcroPlotPrinter.PasswordUser	String	When Allow128Bit is set to 1 or 2 this will set the User password to the text you entered. This is optional and if an empty string then the users will not have to enter a password to open the PDF but the security permissions will still be set.
AcroPlotPrinter.PDFFormat		PDF Format to use
AcroPlotPrinter.Resolution	Long	Output Resolution of the PDF. Available values are: 100, 150, 300, 400, 600, 1200
AcroPlotPrinter.Subject	String	Specifies the subject for the PDF Document Properties
AcroPlotPrinter.Title	String	Specifies the title for the PDF Document Properties
AcroplotPrinter.Watermark	Integer or Long (True/False)	-1=Add a watermark 0=Do not add a watermark
AcroPlotPrinter.WatermarkFont	Integer or Long	Font to use for the Watermark if it is a text watermark 1=Arial 2=Arial Bold 3=Arial Bold and Italic 4=Arial Italic 5=Courier 6=Courier Bold 7=Courier Bold and Italic 8=Courier Italic 9=Times Roman 10=Times Roman Bold 11=Times Roman Bold and Italic 12=Times Roman Italic
AcroplotPrinter.WatermarkFontAngle	Integer or Long	Angle from 0 to 360 degrees

AcroplotPrinter.WatermarkFontColor	Long	Long OLE color value for the color
AcroplotPrinter.WatermarkFontHorizPos	Integer or Long	Distance to offset the watermark in the horizontal direction. Distance is a long value expressed in 1/10th of a mm. Example:1 inch would be a value of 254
AcroplotPrinter.WatermarkFontSize	Integer or Long	Size of font in points
AcroplotPrinter.WatermarkFontVertPos	Integer or Long	Distance to offset the watermark in the vertical direction. Distance is a long value expressed in 1/10th of a mm. Example:1 inch would be a value of 254
AcroplotPrinter.WatermarkFontX	Integer or Long	Horizonatal justification for the watermark 0=Left 1=Centered 2=Right
AcroplotPrinter.WatermarkFontY	Integer or Long	Vertical justification for the watermak 0=Bottom1 =Middle 2=Top
AcroplotPrinter.WatermarkOnTop	Integer or Long (True/False)	-1=Place the watermark on the top 0=Place the watermark below all objects
AcroplotPrinter.WatermarkOpacity	Double	Opacity (transparency) for the text or image 0.0 (invisible) to 100.0 (fully visible) Note: Not allowed with PDF/A-1a and PDF-A-1b formats
AcroplotPrinter.WatermarkText	String	Text string to use for the watermark or the full filename if you want to use an image file for the watermark. Image formats currently supported are bmp and gif.

The Following API Calls would normally not be used but are included for your reference:

AcroPlotAbout(ByVal oAcadApp as Object)
This method will display the AcroPlot About dialog box.

AcroPlotAboutJunior(ByVal oAcadApp as Object)

This method will display the AcroPlot Junior About dialog box

AcroPlotPurchase()

This method opens the AcroPlot Purchase page in your default web browser.

AcroPlotSetup(ByVal oAcadApp as Object)

This method will display the AcroPlot Main Settings dialog.

EmailSales()

This method opens an email to sales@cadzation.com in your default email program.

EmailSalesJunior()

This method opens an email to sales@cadzation.com in your default email program

EmailSupport()

This method opens an email to support@cadzation.com in your default email program

EmailSupportJunior()

This method opens an email to support[@cadzation.com](mailto:support@cadzation.com) in your default email program

GotoWebsite()

This method opens the CADzation website in your default web browser.

OpenHelp()

This method opens the AcroPlot help file

OpenHelpJunior()

This method opens the AcroPlot help file

AcroPlot Advanced API

To create or convert files using the command line mode API or ActiveX automation mode interface, it is required to purchase a license of AcroPlot Auto for each server running the process. If there will be additional PC's running AcroPlot Suite licenses or any of its components, then it is also required to purchase the number licenses needed in addition to the AcroPlot Auto server license.

Notes:

1. If the sOutputFilename exists the PDF files will be merged to it.

2. If the sOutputFilename is read-only or is opened in Adobe Acrobat or Adobe Reader the conversion will fail.

3. If the RestoreSettings is not called or provided a proper settings name then the last used settings in AcroPlot Jr. will be used.

4. If you are merging multiple layouts or drawings into the same PDF make sure that you turn off the OpenPDF setting by using:

SetSetting "AcroPlotSettings.OpenPDF" 0

The Basic API Calls are:

Please see the AcroPlot Standard API Section for the basic API calls and how to initial the AcroPlot.AcroPlotApplication interface.

The Advacned API Calls for AcroPlot Auto customers or customers with a 25 or more AcroPlot Pro license are:

AcadApp() As Object

This property gets or sets the AutoCAD Application object to use for the conversion. This is a standard COM generic object type.

To set the current AutoCAD Application for AcroPlot to use for conversion.
Dim oAcadObject As Object
Dim oAcroPlotApp As Object
Set oAcroPlotApp = CreateObject("AcroPlot.AcroPlotApplication")
Set oAcadObject = CreateObject("AutoCAD.AcroPlotApplication")
Set oAcroPlotApp.AcadApp = oAcadObject
To get the current AcadApp object from AcroPlot
Set oAcadObject = oAcroPlotApp.AcadApp

AcadDoc() As Object

This property gets or sets the AutoCAD Document object to use for the conversion. This is a standard COM generic object type.

StartAutoCAD() As Object

This method starts the version of AutoCAD as defined by the ConvertWith value and returns the standard COM object for the AutoCAD.Application if it succeeds. If that fails then it tries to start the last used version of AutoCAD and returns the AutoCAD.Application if it succeeds. Otherwise returns Nothing.

SetSetting("AcroPlotSettings.ConvertWith", ByVal sAutoCADProgram As string)

This is a special case of the SetSetting standard API call which allows users converting files outside of AutoCAD with the Convert function to specify the version or flavor of AutoCAD for us to use. This would be be the string value as shown in the AcroPlot interface like "AutoCAD 2011" or "AutoCAD 2012 - English". This changes based on versions, flavors, and languages of AutoCAD so it's best to start AcroPlot and look for the values in the Convert With drop down.

SetDWGConvertFrom(ByVal vConvertFrom As Variant) As Long

This method specifies what to convert from the DWG file when you use the Convert method.

Standard input would be a long value based on what you want to convert.

0 = Current layout either modelspace or paperspace

1 = Modelspace

2 = Current Paperspace layout

3 = All Paperspace layouts in order

4 = All Modelspace views

5 = All Paperspace views

SetUseDWGViewer(ByVal bUseViewer As Boolean) As Long

Set to True if you want to use our direct conversion routines instead of AutoCAD for the DWG conversion.

Convert(ByVal *sInputFilename* As String, _
       ByVal *sOutputFilename* As String,
       Optional ByVal *bModal* As Boolean = False,
       Optional ByVal *lOutputformat* As Long = 0,
       Optional ByVal *bTerminateAtEnd* As Boolean = False) As Boolean

This method converts the file to the specified output either by the extension on the *sOutputFilename* or by the *lOutputFormat* specified.

sInputFilename is the full filename of the file that you want to convert. Accepts all supported input formats.

sOutputFilename is the full filename of the file that you want to create.

bModal is an optional parameter if you want the conversion to be modal or modeless. If modal (True) it will not return control to your application until the conversion is finished. If modeless (False) it will return control to your application almost immediately and you will be responsible for watching for the conversion to finish before starting another one.

lOutputFormat is the output format which you would like to convert to. Typically this can be 0 and we will determine the format based off from the file extension of the *sOutputFilename*. But in the case of tiff files where it could be either monochrome or color you would want to specifically tell us which format to use.

bTerminateAtEnd - If set to True this will close the AutoCAD and AcroPlot application and release it at the end of the conversion. It is not recommended to do it this way with some programming languages because an error may occur since the object reference might be lost to the AcroPlot.AcroPlotApplication. Please use the **Quit()** method instead.

The values for the various output formats are:

  pdf = 1

  dwf6 = 2

 'reserved for future use

  BMP = 4

  CAL = 5

  GIF = 6

  gp4 = 7

JPG = 8
PNG = 9
tif = 10 (color tiff at 100 dpi)
tifmono = 11 (monochrome group 3 fax format tiff at 100 dpi)
tifmono200 = 12 (monochrome group 3 fax format tiff at 200 dpi)
tifmono300 = 13 (monochrome group 3 fax format tiff at 300 dpi)
tifmono400 = 14 (monochrome group 3 fax format tiff at 400 dpi)
tif200 = 15 (color tiff at 200 dpi)
tif300 = 16   'not supported reserved for future use
tif400 = 17   'not supported reserved for future use
dwfpdf = 18
dwfpdftif = 19
dwfpdftifmono = 20
dwfpdftifmono200 = 21
dwfpdftifmono300 = 22
dwfpdftifmono400 = 23
pdftif = 24
pdftifmono = 25
pdftifmono200 = 26
pdftifmono300 = 27
pdftifmono400 = 28
'reserved for future use
rasterpdf100 = 30   'not supported reserved for future use
rasterpdf150 = 31   'not supported reserved for future use
rasterpdf200 = 32   'not supported reserved for future use
rasterpdf300 = 33   'not supported reserved for future use
rasterpdf400 = 34   'not supported reserved for future use
rasterpdfmono100 = 35   'not supported reserved for future use
rasterpdfmono150 = 36   'not supported reserved for future use
rasterpdfmono200 = 37   'not supported reserved for future use
rasterpdfmono300 = 38   'not supported reserved for future use
rasterpdfmono400 = 39   'not supported reserved for future use
tif600 = 40   'not supported reserved for future use
tifmono600 = 41   (monochrome group 3 fax format tiff at 600 dpi) 'not supported reserved for future use
dwfpdftifmono600 = 42   'not supported reserved for future use
pdftifmono600 = 43   'not supported reserved for future use
rasterpdf600 = 44   'not supported reserved for future use
rasterpdfmono600 = 45   'not supported reserved for future use
cal200 = 46   'not supported reserved for future use
cal400 = 47   'not supported reserved for future use
cal600 = 48   'not supported reserved for future use
gp4200 = 49   'not supported reserved for future use
gp4400 = 50   'not supported reserved for future use
gp4600 = 51   'not supported reserved for future use
cal300 = 52   'not supported reserved for future use
gp4300 = 53   'not supported reserved for future use
dwfx = 54   'not supported reserved for future use

GetFilesCreated() As Variant

This method returns a string array of the files created during the conversion.

GetFilesCreatedAsString() As String

This method returns a string of the files created during the conversion separated by a semicolon.

GetAPIError() As String

This method returns the error string if an error occurs during the current Convert call. This is the same as checking the registry for the value at HKEY_CURRENT_USER\Software\CADzation\ACROPLOT\API.Error.

EndAutoCAD()

This method ends the instance of AutoCAD that is defined by the AcadApp in AcroPlot and properly releases all objects. This is useful if you want to close AutoCAD after say 100 or 200 conversions to clean up AutoCAD's own memory usage.

Quit()

This method properly quits the AcroPlot.Application object and releases all references. It is recommended that you restart AutoCAD every hundred drawings of so if you are batching with the Convert Method.

Registry Setting to Watch for Conversion Status

If you call the Convert method with the bModal=False then we will return control to your program immediately. In this case you would want to put a loop in your code to watch for use to write the value to the registry that we have finished the conversion. A value of 1 means that the conversion has finished.
The registry key where we wri
HKEY_CURRENT_USER\Software\CADzation\ACROPLOT\API.ConversionFinished (REG_SZ) 0=Working, 1=Finished Successfully, -1=Finished but failed to create output file.

There are 3 other values that we write to the registry that might also be helpful for those using the Convert Method.
HKEY_CURRENT_USER\Software\CADzation\ACROPLOT\API.Error (REG_SZ)
HKEY_CURRENT_USER\Software\CADzation\ACROPLOT\API.Hwnd_AutoCAD (REG_SZ)
HKEY_CURRENT_USER\Software\CADzation\ACROPLOT\API.ResultFilename   (REG_SZ)

AcroPlot Restricted API

The following API calls are restricted and are reserved for use by CADzation only. Although they will show up in the COM Viewer you will not be able to use API methods in your own programs.

The Restricted API Calls are:

AcadDoc_SetDWGProps
AcadDoc_SetDWGPropsBookmarks
AcroPlotJ
Convert2
FilePropertyAuthor
FilePropertyComments
FilePropertySubject
FilePropertyTitle
GetAutoCADLayouts
GetAutoCADViews
IsAuto
OpenAcadDoc
PDFOut
PDFUpdateFiles
RunSiteSpecific
SetCreateNextAsThumbnail
SetMonitor
ShowMonitor
Valid

AcroPlot Advanced Command Line

AcroPlot Command Line access requires an AcroPlot Auto license and is designed to tie into document management systems as a conversion engine that is easily implemented with custom scripting or batch routines.

To create or convert files using the command line mode API or ActiveX automation mode interface, it is required to purchase a license of AcroPlot Auto for each server running the process. If there will be additional PC's running AcroPlot Suite licenses or any of its components, then it is also required to purchase the number licenses needed in addition to the AcroPlot Auto server license.

There are several command line switches that can be passes

/i=   Input filename for the AutoCAD drawing. Filename should be wrapped with quotation marks.

/o=   Output filename or folder name.Filename should be wrapped with quotation marks.
    This should be the full pathname but there are a few shotcuts that can be used.
    If you want to create a PDF with the same name as the input then do not specify
    this and instead set the /f Output format value
   NOTE: If you specify an output file that already exists for PDF and DWF files the file will be appended to the end of the existing file.

/b=   Create Individual Files
    0 = Create a single file - Default
    1 = Create one file per layout

/s=   Settings - Saved Setting name to use from AcroPlot Pro

/t=   Type to plot from
    0 = Current layout either modelspace or paperspace
    1 = Modelspace
    2 = Current Paperspace layout
    3 = All Paperspace layouts in order

/v=   Viewer - Use the viewer for plotting
    0 = Use AutoCAD
    1 = Use the internal viewer
    2 = Use AutoCAD but leave open (Not implemented yet)

/n=   No wait - Does not wait for the conversion to finish and returns
    control back to the command line so it does not tie up the
    program calling it.
    0 = Wait for the conversion to finish
    1 = Do not wait for the conversion to finish. User will check the registry settings to find out when the conversion is finished.

/h=   Creates a prompts to see what values are being passed in. Used for testing only
    1 = Stop executions and display message box for debugging

/f=   AcroPlot Output Format - Needed for file extensions like tif which could have
       multiple different types of output formats
       NOTE: NOT ALL OUTPUT FORMATS WORK YET FROM THE COMMAND LINE
       pdf = 1
       dwf6 = 2
       dwf5 = 3 'not supported any longer
       bmp = 4
       cal = 5
       gif = 6
       gp4 = 7
       jpg = 8
       png = 9
       tif = 10
       tifmono = 11
       tifmono200 = 12
       tifmono300 = 13
       tifmono400 = 14
       tif200 = 15
       tif300 = 16   'not supported reserved for future use
       tif400 = 17   'not supported reserved for future use
       dwfpdf = 18
       dwfpdftif = 19
       dwfpdftifmono = 20
       dwfpdftifmono200 = 21
       dwfpdftifmono300 = 22
       dwfpdftifmono400 = 23
       pdftif = 24
       pdftifmono = 25
       pdftifmono200 = 26
       pdftifmono300 = 27
       pdftifmono400 = 28
       'reserved
       rasterpdf100 = 30
       rasterpdf150 = 31
       rasterpdf200 = 32
       rasterpdf300 = 33
       rasterpdf400 = 34
       rasterpdfmono100 = 35
       rasterpdfmono150 = 36
       rasterpdfmono200 = 37
       rasterpdfmono300 = 38
       rasterpdfmono400 = 39
       unknown = -100
       NOTE: NOT ALL OUTPUT FORMATS WORK YET FROM THE COMMAND LINE

/d= Debug Log Filename Creates a debug log file for the conversion /d="C:\SomeDir\SomeLogName.txt"
       NOTE: Any existing file with this name is deleted each time you run the AcroPlot.exe

/z= Set an Setting that is not directly supported Example /z="AcroPlotPrinter.Grayscale=1" to create for a monochrome PDF.
    NOTE: For more on the various settings please see the [AcroPlot Standard API](#) documentation.

/e= Error Log Filename - Creates an error log file for any errors
    NOTE: The error log is never deleted. Any new errors will be appended to the end of the log file. If you want to watch for an error file to be created either use a unique name each time or delete out the existing error file with your own code prior to running the AcroPlot.exe
    This file is also a semicolon separated file which you could bring into Excel to analyze

/a= Do not show the AcroPlot Monitor
    0 = Do not show the AcroPlot Monitor (monitor still runs in background)
    1 = Show the AcroPlot Monitor
    2 = Do not show the AcroPlot Monitor and close when finished
    3 = Show the AcroPlot Monitor and close when finished

Example of converting DWG to PDF.
AcroPlot.exe /i="C:\File.dwg" /o="C:\File.pdf" /s="DWGToPDF" /b=0 /t=3
In this example we are using the setting named DWGToPDF that we had already defined in AcroPlot Pro. By passing the /b=0 command line switch it will merge all layouts into a single PDF named C:\File.pdf. Also the /t=3 tells us to convert all the paperspace layouts.

Example of converting 3 DWG files to a single PDF.
AcroPlot.exe /i="C:\File1.dwg" /o="C:\File.pdf" /s="DWGToPDF" /b=0 /t=3
AcroPlot.exe /i="C:\File2.dwg" /o="C:\File.pdf" /s="DWGToPDF" /b=0 /t=3
AcroPlot.exe /i="C:\File3.dwg" /o="C:\File.pdf" /s="DWGToPDF" /b=0 /t=3
In this case you would want to make sure that the output pdf file does not exist. Then the first DWG will be created as the output file. The next two since you are passing the same output filename to us we will append those onto the existing PDF so you will end up with a single PDF of the 3 drawings.

Also in this example we are using the setting named DWGToPDF that we had already defined in AcroPlot Pro. By passing the /b=0 command line switch it will merge all layouts into a single PDF named C:\File.pdf. Also the /t=3 tells us to convert all the paperspace layouts.

Example of Merging 2.PDF to 1.PDF
AcroPlot.exe /i="C:\2.pdf" /o="C:\1.pdf"

Example of Merging 1.pdf, 2.pdf, and 3.pdf together to a new filename
AcroPlot.exe /i="C:\1.pdf" /o="C:\NewFilename.pdf"
AcroPlot.exe /i="C:\2.pdf" /o="C:\NewFilename.pdf"
AcroPlot.exe /i="C:\3.pdf" /o="C:\NewFilename.pdf"

Example of Adding a Text Watermark to a PDF
NOTE: You must specify a new filename otherwise it will try to append the file to itself.

AcroPlot.exe /i="C:\1.pdf" /o="C:\WatermarkedFile.pdf" /s="SavedSettingsForWatermark"

or

AcroPlot.exe /i="C:\1.pdf" /o="C:\WatermarkedFile.pdf" /z="AcroPlotPrinter.Allow128Bit=1" / z="AcroPlotPrinter.Watermark=1" /z="AcroPlotPrinter.WatermarkFont=2" /z="AcroPlotPrinter.WatermarkFontAngle=0" /z="AcroPlotPrinter.WatermarkFontColor=255" /z="AcroPlotPrinter.WatermarkFontHorizPos=1" /z="AcroPlotPrinter.WatermarkFontSize=12" /z="AcroPlotPrinter.WatermarkFontVertPos=2" / z="AcroPlotPrinter.WatermarkFontX=0" /z="AcroPlotPrinter.WatermarkFontY=254" /z="AcroPlotPrinter.WatermarkOnTop=1" / z="AcroPlotPrinter.WatermarkText=||PAGE|| of ||PAGETOTAL||"

The above sample would located a red text watermark centered horizontally and 1 inch (25.4 mm) above the bottom of the page using a font size of 12 points. For more on the various settings please see the [AcroPlot Standard API](#) documentation.

Example of Adding an Image Watermark to a PDF
AcroPlot.exe /i="C:\1.pdf" /o="C:\WatermarkedFile.pdf" /s="SavedSettingsForImageWatermark"

or

AcroPlot.exe /i="C:\1.pdf" /o="C:\WatermarkedFile.pdf" /z="AcroPlotPrinter.Allow128Bit=1" /z="AcroPlotPrinter.Watermark=1" / /z="AcroPlotPrinter.WatermarkFontAngle=0" /z="AcroPlotPrinter.WatermarkFontHorizPos=1" /z="AcroPlotPrinter.WatermarkFontVertPos=2" /z="AcroPlotPrinter.WatermarkFontX=0" / z="AcroPlotPrinter.WatermarkFontY=254" /z="AcroPlotPrinter.WatermarkOnTop=1" /z="AcroPlotPrinter.WatermarkText=C: \SomeImageFile.gif"

The above sample would located the gif image centered horizontally and 1 inch (25.4 mm) above the bottom of the page using a font size of 12 points. For more on the various settings please see the [AcroPlot Standard API](#) documentation.

Example of Adding Password Protection to a PDF
AcroPlot.exe /i="C:\1.pdf" /o="C:\PasswordProtectedFile.pdf" /s="SavedSettingsWithPassword"

or

AcroPlot.exe /i="C:\1.pdf" /o="C:\PasswordProtectedFile.pdf" /z="AcroPlotPrinter.Allow128Bit=2" /z="AcroPlotPrinter.PasswordAuthor=AuthorPassword" /z="AcroPlotPrinter.PasswordUser=UserPassword" /z="AcroPlotPrinter.Allow128Assembly=-1" /z="AcroPlotPrinter.Allow128BitExtract=0" /z="AcroPlotPrinter.Allow128BitInteractiveForms=-1" /z="AcroPlotPrinter.Allow128BitPrintLow=-1" /z="AcroPlotPrinter.AllowChanges=0" /z="AcroPlotPrinter.AllowCopy=-1" /z="AcroPlotPrinter.AllowNotes=-1" z="AcroPlotPrinter.AllowPrint=-1"

API Sample - VBA

This sample will create 2 different files of the current layout. The first one will be named after the drawing and will use the AcroPlot Jr. defaults. The second one will be in Monochrome, 11x17, Scaled-To-Fit and will be named after the drawing and Tabloid-Monochrome.

Sample VB/VBA Application Using the AcroPlotJ Methods

```vb
Sub TestCurrentLayout()
   On Error Resume Next
   Dim oAcroPlotApp As Object
   Dim oAcadApp As AcadApplication
   Dim sSettingName As String
   Dim sOutputFile As String
  'First try to get the AcroPlot Application Interface
  'We'll use late binding to make it easier
   Set oAcroPlotApp = CreateObject("AcroPlot.AcroPlotApplication")
   If oAcroPlotApp Is Nothing Then
      MsgBox "Unable to get the AcroPlot Application Interface"
      Exit Sub
   End If
  'Get the current AutoCAD.Application Object
   Set oAcadApp = ThisDrawing.Application
  'Set the Settings Name use "AcroPlotJunior." for the last used settings in AcroPlot Jr.
   sSettingName = "AcroPlotJunior"
  'Set the output filename equal to the drawing name .pdf
   sOutputFile = ThisDrawing.FullName
   sOutputFile = Strings.Left$(sOutputFile, Len(sOutputFile) - 4) & ".pdf"
  'Now we should set the Undo mark because if the layout was set to a printer that does
  'not exist we cannot restore that due to AutoCAD limitations.  But the Undo can.
   ThisDrawing.StartUndoMark
   'Now let convert the current layout
   oAcroPlotApp.Init
 '***************VERY IMPORTANT***************
  'Provide the AcroPlot Auto unlock code here so the filename dialog does not popup.
  'Other users will have the filename dialog popup per our standard licensing for the AcroPlot Suite, and will need to
select the "Save As" button for each file created.
  '***************VERY IMPORTANT***************
   oAcroPlotApp.UnlockAPI "YOUR_REGISTRATION_CODE_FOR_ACROPLOT_AUTO"

   oAcroPlotApp.RestoreSettings sSettingName
   oAcroPlotApp.SetOutputFilename sOutputFile
   oAcroPlotApp.AcroPlotJCurrent oAcadApp
  'Now let's change the papersize to tablod, monochrome, and scaled to fit
   sOutputFile = ThisDrawing.FullName
   sOutputFile = Strings.Left$(sOutputFile, Len(sOutputFile) - 4) & "-Tabloid-Monochrome" & ".pdf"
   oAcroPlotApp.SetOutputFilename sOutputFile
   oAcroPlotApp.SetSetting "DwgPaperSize", 3
   oAcroPlotApp.SetSetting "Grayscale", 1
   oAcroPlotApp.SetSetting "PlotScale", 0
```

```
      Err.Clear
      oAcroPlotApp.AcroPlotJCurrent oAcadApp
     'Release the AcroPlot Application
      oAcroPlotApp.Quit
      Set oAcroPlotApp = Nothing
     'Now let's undo the changes
      ThisDrawing.EndUndoMark
      ThisDrawing.SendCommand "_u "
   End Sub
```

Sample VB/VBA Application Using the Convert Method

To create or convert files using the command line mode API or ActiveX automation mode interface, it is required to purchase a license of AcroPlot Auto for each server running the process. If there will be additional PC's running AcroPlot Suite licenses or any of its components, then it is also required to purchase the number licenses needed in addition to the AcroPlot Auto server license.

```
Private Sub ConvertFile(ByVal sInputFile As String, ByVal sOutputFile As String, ByVal sSettingName As String)
 On Error Resume Next
 Dim oAcadObject As Object
 Dim oAcroPlotApp As Object
 Dim sFilesCreated As String
 Dim bResults As Boolean
 Dim lWait As Long
 Dim lFilesOpened As Long
 Dim lLoop As Long
 Dim sConversionFinished As String
 Set oAcroPlotApp = CreateObject("AcroPlot.AcroPlotApplication")
     If oAcroPlotApp Is Nothing Then
             MsgBox "Unable to get the AcroPlot Application Interface"
             Exit Sub
       End If
 oAcroPlotApp.RestoreSettings sSettingName
 'Now maybe we want to override the settings and make the output grayscale
 oAcroPlotApp.SetSetting "Grayscale", 1
 'Since we do not want to append to the end of the existing file we should delete it first if it exists.  There should of coarse be
 some error checking here.
 Kill sOutputFile
 'Lets use our own internal converter for this first one.
 oAcroPlotApp.SetSetting "UseInternalViewerForPlotting", -1
 'Now we want to do the conversion modal so it will not return control to our application until it finishes.
 Call oAcroPlotApp.Convert(sInputFile, sOutputFile, True, 0, False)
 sFilesCreated = oAcroPlotApp.GetFilesCreatedAsString

 'For the next ones let's use AutoCAD for the conversion.
 oAcroPlotApp.SetSetting "UseInternalViewerForPlotting", 0
 'Now let's say we want to test running it 200 times and having it close AutoCAD after 100 files
```

```vb
    'We will also do this conversion modeless so we would need to have a loop to wait and check every second if the conversion is finished.
 For lLoop = 0 to 199
     'Since we do not want to append to the end of the existing file we should delete it first if it exists.  There should of coarse be some error checking here.
     Kill sOutputFile
     Call oAcroPlotApp.Convert(sInputFile,sOutputFile, False, 0, False)
     'Here now we want to wait until we write a value other than 0 to the HKEY_CURRENT_USER\Software\CADzation\ACROPLOT\API.ConversionFinished
     'The ReadRegVal would be a wrapped function in VB or VBA to read a registry value
     For lWait = 0 to 300
         Sleep 1000 'Windows API function to wait specified number of milliseconds
         sConversionFinished = ReadRegVal(HKEY_CURRENT_USER, "Software\CADzation\ACROPLOT", "API.ConversionFinished")
         If sConversionFinished = "1" then
             'Created the output file
             Exit For
         Else if sConversionFinished = "-1"
             'Failed to create the output file
             Exit For
         End if
     Next
     If lFilesOpened < 100 then
         lFilesOpened = lFilesOpened + 1
     Else
         lFilesOpened = 0
         oAcroPlotApp.EndAutoCAD
     End if
 Next
 'Now let's properly exit AcroPlot.Application
 oAcroPlotApp.Quit
 Set oAcroPlotApp = Nothing
End Sub
```

Online API Samples

We also have a few sample projects created online that you can download and look at.

The samples are located in a zip file at:
https://www.cadzation.com/downloads/AcroPlotAPISamples.zip

Currently there is a VB6 sample and a VS 2005 ASP.Net/C# sample in the zip.

How To Tell If AcroPlot Is Running

In certain cases you may want to disable certain scripts from running that may halt the batch processing in AutoCAD if AcroPlot or AcroPlot Pro are running. We have provided 4 registry settings that save the time that AcroPlot was started and also the time that we tried to start opening the current AutoCAD drawing to help make it easy for you to determine if AcroPlot is doing the processing.

The values are stored as doubles in the Registry and the decimal part of the value is the percentage of the day. So multiplying the decimal portion by 86400.0 will give you the number of seconds elapsed since midnight. The integer portion changes between the standard Windows date and the AutoCAD Jullian date by 2415019 days.

The values are saved under the current user's values:
HKEY_CURRENT_USER\Software\CADzation\AcroPlot

For VB users you can use the following 2 keys which are based on the system date and time given by the Now function.
StartTime - Stores the date and time that the AcroPlot dialog was started. This value is reset to 0 when the form is closed so in theory yo could just check this value as long as the program is not ended from the task manager or by some other abnormal way.
Open Time - Stores the date and time that AcroPlot tried to open the last drawing in AutoCAD. This is the safest value to use as you could determine the maximum amount of time it takes to open your drawings and then compare the times. This value is also reset to 0 when the form is closed.

For AutoLisp and VisualLisp users you can use the following 2 keys (using the Visual Lisp vl-registry-read) which are based on the value returned from the AutoCAD date system variable.
StartTimeAutoCAD - Stores the date and time that the AcroPlot dialog was started. This value is reset to 0 when the form is closed so in theory yo could just check this value as long as the program is not ended from the task manager or by some other abnormal way.
Open TimeAutoCAD - Stores the date and time that AcroPlot tried to open the last drawing in AutoCAD. This is the safest value to use as you could determing the maximum amount of time it takes to open your drawings and then compare the times. This value is also reset to 0 when the form is closed.

Technical Support

Technical support for AcroPlot Auto is best handled by visiting our online helpdesk at: www.cadzation.com/support.htm

From there you can both search our knowledgebase and also submit a new support request. You can also upload any files that you may be having problems with as long as they are under 6 MB in size.

We have found in the past that pure email support is too unreliable with customer's email servers filtering out our emails at times. The online helpdesk allows you to log in and check the status at any time in case you do not receive our emails.

Copyrights and Trademarks