

HOWTO: Specify a Custom Extraction location for Atalasoft.PDFium.dll (resolve PdfDecoder issue)

Applicability

This article applies to version 11.0 and newer as the PDFium engine has additional DLLs that are auto extracted from Atalasoft.DotImage.PdfReader.dll

If you're running into issues in either a web or console or desktop app where there is an error about temp file access / missing Atalasoft.PDFium.dll, this may be a permissions issue where the process is unable to extract due to insufficient rights. This almost never affects desktop users or other processes running under an interactive domain or user account as the default location is in the user temp directory (%temp%) which will be something along the lines of

```
c:\users\YOUR_USERNAME\AppData\Local\Temp\Atalasoft\DotImage\FULL_VERSION\Atalasoft.PDFium.dll
```

Where **YOUR_USERNAME** is the username you logged in as and **FULL_VERSION** is the specific [full exact version of DotImage](#) that was being used.

If you've used different versions of DotImage over the years, you may find you have multiple versions extracted.

However, for process that run as a system service, the "user temp directory" doesn't apply and those may try to use restricted system temp locations such as C:\Windows\Temp. This is especially problematic for web processes.

This is why it is sometimes helpful to specify an alternate extraction location where you can be sure the web process has full rights.

How To Manually Specify Extraction Location

Controlling where the Atalasoft.PDFium.dll extracts to is sometimes required

This is known behavior. Please see: [INFO: Changes Introduced in DotImage 11.0](#) under Temp File Extraction for Atalasoft.PDFium.dll

The good news is that in newer versions (11.0.0.9 and newer) there is an explicit way to configure where to look..

HOWTO: Specify a Custom Extraction location for Atalasoft.PDFium.dll (resolve PdfDecoder issue)

Check out the PdfDecoder API reference.

To configure PDFium library location, you should add the next line to app.config file:

```
Atalasoft> <PdfDecoder PdfiumLocation="C:\Folder\Subfolder\SubSubFolder"/> <!-- Components sections --> ... </Atalasoft>
```

You can also use a relative path

```
Atalasoft> <PdfDecoder PdfiumLocation="..\Another\Sub\Folder"/> <!-- Components sections --> ... </Atalasoft>
```

You will need to update your web.config to point it at a valid location where it can extract under the web processes' credentials.. the sample here was for app.config.. it needs to be altered for web.config.. this is a general guidance based on

[Atalasoft.Shared.AtalasoftConfigSection](#)

```
?xml version="1.0"?> <configuration> <configSections> <section name="Atalasoft" type="Atalasoft.Shared.AtalasoftConfigSection, Atalasoft.Shared"/> </configSections> <runtime> <gcServer enabled="true"/> </runtime> <Atalasoft> <PdfDecoder PdfiumLocation="C:\SomePath\SimpleWDVwithWDT\SimpleWDVwithWDT\bin"/> </Atalasoft> <system.web> ... removed for brevity here... </system.web> </configuration>
```

Common Causes of Errors

Permissions Issues

Failure to Extract

The symptom of a permissions issue where the process is completely unable to extract the Atalasoft.PDFium.dll would be that no PDFs are rendered despite having proper code references and licensing. You can verify this by determining the %temp% location for the user running the process, and digging down to see if the Atalasoft\DotImage\FULL_VERSION\Atalasoft.PDFium.dll is present.

The error/stack trace may also say it was unable to find Atalasoft.PDFium.dll and may helpfully also list locations where it looked. This can be useful in situations where you're not

HOWTO: Specify a Custom Extraction location for Atalasoft.PDFium.dll (resolve PdfDecoder issue)

entirely sure which user/account is running the process.

You may also find that [ProcMon](#) is helpful in troubleshooting by looking to see where it is failing to find the DLL.

The key that makes Failure to Extract a Permissions issue is when the process did not have write access to create the directories / extract the Atalasoft.PDFium.dll at all.

Failure to Execute

The symptom of Failure to execute is that the Atalasoft.PDFium.dll is present exactly where it's expected to be....meaning that the version folder [matches the Product Version](#) of the Atalasoft.dotImage.PdfReader.dll being run.

The symptom is most commonly that PDFs error out read, but if you look in the appropriate %temp% location for the process, the folders and DLLs are present. You may need to dig into the file permissions to see if the web process actually has rights.

If you have some files /processes reading and some not, it may be that there is some kind of impersonation being used either in the web process, or the desktop process attempting to use the PdfDecoder. Stack traces and/or [ProcMon](#) logs can help sort out if this is your cause.

The key here is that the Atalasoft.PDFium.dll exists but that the user running the process can't execute it.

Temp File Cleanup

It is possible that if your server does intermittent temp folder cleanups (either on a schedule or due to a manually triggered run of Windows Disk Cleanup, etc. If you have a running web process that already successfully extracted the Atalasoft.PDFium.dll, that such a cleanup is "pulling the rug out from under it". This can end up looking like one of the previous permissions issues depending on whether you catch the situation with the DLL present or not (where in the Cleanup and execution cycle you go looking).

NOTE that Atalasoft.dotImage.PdfReader.dll extracts Atalasoft.PDFium.dll to the folder based on its version but does not delete it. Thus if you had old versions present before and they're gone this is a hint that cleanup has been run.

HOWTO: Specify a Custom Extraction location for Atalasoft.PDFium.dll (resolve PdfDecoder issue)

The good news is that restarting the process generally fixes this because on startup it will determine if the DLL needs extraction and take care of it.

Its when the "rug is pulled out from under it" that you get the crashing from this scenario. This is because it doesn't constantly check- it checks/extracts on first instantiation.

If you have persistent cleanup related issues choosing an exaction location that is not subject to automatic / periodic cleanup may resolve your issue.

Last Update

2026-10-19 - TD - (Normalized "PDFium")

Atalasoft Knowledge Base

<https://www.atalasoft.com/kb2/KB/50412/HOWTO-Specify-a-Custom-Extraction-lo...>