

FAQ: Why Do Some PDFs Throw Exceptions in Some Atalasoft Classes but Read in Acrobat Reader Just Fine

READING / VIEWING

Many of Atalasoft's customers make extensive use of Atalasoft's PDF capabilities. We have several classes for dealing with PDFs .. the simplest of which is PdfDecoder.. this allows PDF format files to have their pages rendered in our viewers and for conversion to other formats and/or Image processing.

This component "rasterizes" meaning that it essentially returns an image snapshot of what a given page in a PDF looks like when rendered.

It's important to understand that this action .. much like opening a PDF in Acrobat Reader or an other PDF reader tool: it only needs to make a best attempt to render/view.. If the PDF is corrupt/damaged, some attempt may be made to make a "best guess" in order to render the content.

MODIFYING

However, when you are attempting to open a PDF in one of our classes that needs to modify the PDF (PdfDocument, PdfGeneratedDocument, PdfAnnotationDataImporter, PdfAnnotationDataExporter), then Atalasoft will throw an exception if it encounters errors rather than continuing.

Atalasoft offers an ability to attempt repairs. Explicitly, you can call PdfDocument.Repair(...) on a given PDF to attempt a direct repair. You can control the scope of repairs by adjusting properties in the RepairOptions you send in (see the API reference included with our SDK for details on PdfDocument.Repair(...) and its various overloads).

The repair is needed because PdfDocument and certain other Atalasoft classes like PdfAnnotationDataExporter need to modify the PDF contents.. if one attempts to modify a damaged PDF one could utterly destroy the PDF structure.. making it worse

Our components are set up to "do no harm" which is why repair is something you must either do manually with PdfDocument.Repair or via adding RepairOptions to your PdfDocument constructor.. you're explicitly saying you are OK with modifying the document to attempt a repair.

FAQ: Why Do Some PDFs Throw Exceptions in Some Atalasoft Classes but Read in Acrobat Reader Just Fine

EXPLANATION

Bottom line, our PdfDocument, PdfGeneratedDocument, PdfAnnotationDataExporter and PdfAnnotationDataImporter classes must have a valid document that is error free to work with as they need to MODIFY the PDF.. and if one attempts to modify a broken PDF, one can make it even worse.. possibly unrecoverable.

This is why Acrobat Reader can often read your file even when our PdfDocument or other components throw a PdfException: because Reader and our own PdfDecoder only have to read the PDF, and can make attempts at guessing at certain missing data (unless the PDF is too corrupt).

In short, you must understand that there are an entirely different set of restrictions/rules at play when you need to merely view the PDF versus making changes to the structure of the PDF, and we cannot perform modifications on a corrupt PDF as we could and would cause more damage. Our repair tools are able to help, but by using them you are implicitly acknowledging that you are now working with "damaged goods" and making an attempt to repair it

Some PDFs are going to be too corrupt to repair and some PDFs may repair in unexpected ways... for example:

A common case is "orphaned pages" .. where a PDF has got page objects in it structure that are not actually displaying/tied to pages in the document.

Our PdfDocument.Repair cannot tell which pages were intentionally orphaned:

Did the creator mean to delete the page, but only delete the reference without removing the object?

or did they intend to add a page but only added the page object without actually adding the required reference?

There is no way for us to tell what was "in the intent of the creator of the PDF" so we just do our best to recover the PDF into a non-corrupt state so that it can be safely opened in classes that can modify it. Some customers may not like that suddenly, these extra pages they did not know existed "pop up into the PDF" but the repair is merely finding objects that exist in the page structure that are not properly linked in to display and connecting them.

FAQ: Why Do Some PDFs Throw Exceptions in Some Atalasoft Classes but Read in Acrobat Reader Just Fine

SUMMARY

When you use PdfDocument.Repair(...) or one of our class constructors that take a RepairOptions instance to allow for automatic repair of PDFs, you can use the properties of the RepairOptions to control the scope of repair, but by using the repair, you're implicitly acknowledging that this is a best attempt to repair a corrupt (non-spec-compliant) PDF.. that Repair is by nature a potentially imperfect process. Depending on your setting for RepairOptions.MaximumAllowableSeverity and other properties, it will make a best attempt to make the PDF spec-compliant. This can have consequences to the appearance of the PDF.. please see our API reference ApiReference.chm which ships with the SDK(for 11.0 it's found in C:\Program Files (x86)\Atalasoft\DotImage 11.0\Help)

Original Article

Q10463 - FAQ: Why Do Some PDFs Throw Exceptions in Some Atalasoft Classes but Read in Acrobat Reader Just Fine

Last Update

2026-09-17 - TD - (Full Reformatting)

Atalasoft Knowledge Base

<https://www.atalasoft.com/kb2/KB/50010/FAQ-Why-Do-Some-PDFs-Throw-Exception...>