INFO: Changes Introduced in DotImage 11.0

Release Notes

The full official release notes for DotImage can be found below:

Release Notes Page

Individual releases

Initial Release (Aug 2017)- v11.0.0.0.00297

FixPack 1 (Sep 2017) - v11.0.0.1.00338

FixPack 2 (Oct 2017) - v11.0.0.2.00358

FixPack 3 (Nov 2017) - v11.0.0.3.390

FixPack 4 (Dec 2017) - v11.0.0.4.408

FixPack 5 (Jan 2018) - v11.0.0.5.418

FixPack 6 (March 2018) - v11.0.0.6.432

FixPack 7 (April 2018) - v11.0.0.7.458

FixPack 8 (June 2018) - v11.0.0.8.478

FixPack 9 (August 2018) - v11.0.0.9.498

Fixpack 10 (September 2018) - v11.0.0.10.507

FixPack 11 (November 2018) - v11.0.0.11.517

Support has received many requests to make a more "search friendly" version of the release notes.

Your wish is our command:

INFO: 11.0 Full Release Notes

New Feature

WebDocumentThumbnailer Multiple Select (11.0.0.7 and later)

INFO: Changes Introduced in DotImage 11.0

With the release of 11.0.0.7 (April 2018) we have added Multiple Selection to the WebDocumentThumbnailer control

Please see the following KB article for full details:

INFO: WebDocumentThumbnailer Multi Select (New in 11.0.0.7)

Breaking Changes

Tesseract3Engine OCRResources Updated

Applies to v11.0.0.10 AND NEWER

As of FixPack 10 (v11.0.0.10.507), we've updated our Tesseract3Engine OCRResources. We used Tesseract 3.02 previously, but a fix for a SystemAccessViolationException required that we move to 3.04

Please see:

INFO: Tesseract Resources (Tessdata) Updated to 3.04 in 11.0.0.10

WebDocumentViewer Resource Changes

Applies to v11.0.0.1.00338 and newer

WebDocumentViewer resources - updated jQuery and jQueryUI versions

Old File	New File
jquery-1.7.1.min.js	jquery-1.11.0.min.js
jquery-ui-1.8.14.custom.css	jquery-ui-1.10.4.min.css
jquery-ui-1.8.14.custom.min.js	jquery-ui-1.10.4.min.js
clipboard.js	-- removed --

INFO: Changes Introduced in DotImage 11.0

clipboard.min.js	clipboard.min.js (re-added in 11.0.0.2.00358)

This means you will need to update your script references from

<script src="WebDocViewer/jquery-1.7.1.min.js" type="text/javascript"></script>

<script src="WebDocViewer/jquery.easing.1.3.js" type="text/javascript"></script>

<script src="WebDocViewer/jquery-ui-1.8.14.custom.min.js" type="text/javascript"></script>

<script src="WebDocViewer/raphael-min.js" type="text/javascript"></script>

<script src="WebDocViewer/atalaWebDocumentViewer.js" type="text/javascript"></script>

<link href="WebDocViewer/jquery-ui-1.8.14.custom.css" rel="Stylesheet" type="text/css" />

<link href="WebDocViewer/atalaWebDocumentViewer.css" rel="Stylesheet" type="text/css" />

<meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no">

to

<meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no">

Known Issues

Jp2Encoder deprecated

New Jpeg2000Encoder available in 11.0.0.3 and newer

The Jp2Encoder (For encoding Jpeg2000 files and for using Jpeg2000 compression) was discovered to throw an exception when attempting to use less common WavletFilterMethod settings. When engineering analyzed the situation, they realized that fixing it would cause a breaking change for existing Jp2Encoder users and changing several default settings. Therefore, the decision was made to deprecate the Jp2Encoder class but keep it in DotImage... and provide a replacement Jpeg2000Encoder which had the correct properties and settings to allow for less common configuration settings

While you do not need to change your solution to use Jpeg2000Encoder unless you're receiving exceptions on save due to WavletFilterMethod and other advanced settings, it is recommended to use the Jpeg2000Encoder if possible

Also note, the Legacy Demos will be getting updated to use the Jpeg2000Encoder which means that if you are using 11.0.0.2 or earlier you will not be able to compile as the Jpeg2000Encoder class is not present until 11.0.0.3 -you can resolve this issue by changing any instance of Jpeg2000Encoder in the code back to use Jp2Encoder instead

Missing Classic ASP.NET Web Resource Files

FIXED in 11.0.0.2 and newer

Users that use Web Resources folders with older ASP.NET controls (This does NOT apply to WebDocumentViewer or WebCapture) may note that the following files are missing

```
:\Program Files (x86)\Atalasoft\DotImage 11.0\bin\WebResources\ajax-loader_drk.gif C:\Program
Files (x86)\Atalasoft\DotImage 11.0\bin\WebResources\AtalaCanvas.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\Background_Splash.gif C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\ClientTools.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\CommentMinimized.png C:\Program Files
```

```
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\Enums.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\excanvas-compressed.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\mask.png C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\spacer.gif C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebAnnotation.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebAnnotationController.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebAnnotationViewer.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebAnnotationViewerSplash.gif C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebImageViewer.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebImageViewerChild.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebImageViewerSplash.gif C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebThumbnailViewer.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebThumbnailViewerChild.js C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebThumbnailViewerSplash.gif C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\Zoom.cur C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\ZoomIn.cur C:\Program Files
(x86)\Atalasoft\DotImage 11.0\bin\WebResources\ZoomOut.cur
```

This is a known issue and should be addressed in 11.0.0.2 in end of October 2017

In the meantime, we've attached WebResources_11.0.0.1.00338.zip to this case

Temp File Extraction for Atalasoft.PDFium.dll

Atalasoft changed to a new PDF engine in 11.0. Due to internal reasons, we need to extract the Atalasoft.PDFium.dll into the TEMP directory of the current user's profile.

For the vast majority of users, this should be completely transparent. However, if you have a particularly locked down system and receive permissions errors relating to the Atalasoft.PDFium.dll or the temp folders, here is the information you will need to resolve the issue:

For windows users/apps running under "normal" accounts that would be c:\users\YOUR_NAME\AppData\Local\Temp\

(full path would be c:\users\YOUR_NAME\AppData\Local\Temp\Atalasoft\DotImage\VERSION_NUMBER\x86orx64\Atalasoft.PD
)

For services running under system accounts.. including web applications under IIS, the user account that runs it may be a non-interactive account without a profile. In those cases it falls back on

c:\Windows\TEMP\

(full path would be
c:\Windows\TEMP\Atalasoft\DotImage\VERSION_NUMBER\x86orx64\Atalasoft.PDFium.dll)

Once it's extracted for that version, it should be able to read from others, and you may just be able to (depending on how locked down your system is) set read/execute permissions to be a bit more Open.

If you're encountering this in a web app or service, your other option would be to ensure that the user running the IIS process / windows service has an actual user profile of its own and that each app has a unique user.

As of 11.0.0.9, we added configuration options to allow you to specify location to extract to.

For applications with appl.config, the entry is done there along these lines:

<Atalasoft> <PdfDecoder PdfiumLocation="C:\Folder\Subfolder\SubSubFolder"/>

<!-- Components sections --> ... </Atalasoft>

For web apps and services that have a web.config, you need to add the correct sections:

<?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:\someFolder\YourAppFolder\YourAppName\bin" />

</Atalasoft>

<system.web>

... removed for brevity here...

</system.web>

</configuration>

Or, if you're using IIS 7.5, it is possible to configure temp folder for each app

INFO: Changes Introduced in DotImage 11.0

https://serverfault.com/questions/350550/is-it-possible-to-configure-the-location-of-temporary-asp-net-fil

PdfReader Related License Changes

DotImage 11.0 includes an entirely new PdfReader assembly.

Users with existing PRS2 / PRSX2 licenses will be issued new PRP2 / PRPX2 serial numbers automatically.

New purchases of PdfReader add-on will provide PRP2 / PRPX2 serials only.

What this means is that for licensing 11.0 and newer, users will want to activate their new PRP2 / PRPX2 serials instead of their PRS2 / PRSX2 serials.

If you are a currently licensed user of the PdfReader add-on and have existing PRS2 / PRSX2 serials under maintenance, and have not received your new PRP2/PRPX2 serial(s) from Atalasoft, please check with the person within your company who handles the Atalasoft billing/renewals.

If you're unsure of who within your company would have received the updated serials, you may open a support ticket and we can assist.

NOTE: if you need to activate / license a 10.7 or older DotImage with PdfReader, you will need to use your PRS2 / PRSX2 serial(s) as the PRP2/ PRPX2 serials are only for 11.0 and newer.

Support for Large PDF Files > 2 GiB

As a "side effect" of updating to the newer Pdfium engine, we now can support PDF files > 2 GiB in size (previously even on 64 bit systems, the max size for a PDF file was 2 GiB due to the use of signed 32 bit integers for stream position pointers internal in the PdfReader library

This limit is no longer in place, so you should be able to read from/ write to PDF files up to 4 GiB in size

INFO: Changes Introduced in DotImage 11.0

Client-Side (JavaScript) API Documentation Moved

The DevelopersGuide.pdf which comes with DotImage (C:\Program Files (x86)\Atalasoft\DotImage 11.0\Help\DevelopersGuide.pdf) no longer contains client-side JavaScript API documentation for WebDocumentViewer and WebDocumentThumbnailer. that documentation has been moved to

WebDocumentViewer JavaScript API:

https://atalasoft.github.io/web-document-viewer/

Additionally, we are no longer shipping WingScan API Reference.pdf in the WebCapture Resources folder:

C:\Program Files (x86)\Atalasoft\DotImage 11.0\bin\WebResources\WebCapture

That documentation has been moved to:

WebCaptureService (WingScan) JavaScript API:

https://atalasoft.github.io/web-capture-service/

.NET 2.0 and .NET 4.0 DLLs updated to 3.5 and 4.5.2

DotImage has shipped with .NET 2.0 DLLs since version 6.0 and with .NET 4.0 DLLs since version 9.0. Microsoft is dropping support for .NET Framework 2.0 and 3.0 and also for .NET Framework 4.0, 4.5, and 4.5.1.

DotImage in 11.0 and newer will no longer provide .NET 2.0 and .NET 4.0 DLLs, and instead will ship with .NET 3.5 and .NET 4.5.2 respectively. The .NET 4.5.2 DLLs will work with the .NET 4.6 / 4.7 + variants as well.

Support for ASP.NET Core Apps for WebDocumentViewer and WebCapture

We have support for ASP.NET Core targeting .NET Framework for our WebDocumentViewer and WebCapture components.

Please see FAQ: Support for ASP.NET Core / .NET Core for full details.

NOTE: we do not support .NET Core native (full .NET Core) just the ASP.NET Core subset when targeting .NET Framework with our modern web controls. There is no support for ASP.NET Core or .NET Core for our legacy web controls or any other controls.

Annotation Burning in WebDocumentViewer

We've added annotation burning to WebDocumentViewer. to use this feature, you need to set the burn property of a given annotation to "true" and update the annotation. Then on save, the annotation will be burned to the saved document instead of saved in annotation data.

```
/ Burn all annotations on a single page function burnAllAnnosOnPage(pageIndex) { var annos = _viewer.getAnnotationsFromPage(pageIndex); if (annos != null) { for(var i=0; i < annos.length; i++){ var anno = annos[i]; anno.burn = true; anno.update(); } } } // burn all annos on all pages function burnAllAnnos() { var pageCount = _viewer.getDocumentInfo().count; if (pageCount != null && pageCount > 0) { for (var i = 0; i < pageCount; i++) { burnAllAnnosOnPage(i); } } }
```

now, when you call save on the viewer, any annotations with burn=true will be burned to the image

Reloading Individual Pages in WDV/WDT

11.0 introduced the new feature of being able to reload a single changed page in the WebDocumentViewer and WebDocumentThumbnailer controls

If using WDT with WDV, then calling

_thumbs.document.reloadPage(pageIndex);

will reload the thumbnail

_viewer.document.reloadPage(pageIndex);

will reload the Web Document Viewer page

PdfGeneratedDocument (DotPdf) ImportPages and Form Merge Conflict Resolution

PdfGeneratedDocument now has a new ImportPages feature designed to easily allow for importing pages from additional external PDFs.

PdfDocument can not combine PDFs where more than one of the documents has AcroForm elements. The new ImportPages feature of PdfGeneratedDocument includes conflict resolution events to allow for handling such cases

HOWTO: Combine Multiple PDF Files With Fillable Forms (AcroForms)

Quick Example

```
rivate void MergePdfWithFormsConflicResolution(string baseFile, string importFile, string outFile) { using (FileStream baseFileStream = new FileStream(baseFile, FileMode.Open, FileAccess.Read, FileShare.Read)) { using (PdfGeneratedDocument genDoc = new PdfGeneratedDocument(baseFileStream)) { using (FileStream importFileStream = new FileStream(importFile, FileMode.Open, FileAccess.Read, FileShare.Read)) { genDoc.ImportPages(importFileStream, new ImportOptions { //// These are additional options available //InsertIndex = -1, //PagesToImport = new int[] {1,3,5,7}, //OwnerPassword = "", //UserPassword = "", //RepairOptions = new Atalasoft.PdfDoc.Repair.RepairOptions(), //// This option is the key one for merging with acroforms conflict resolution FormFieldsConflictHandler = ResolveFormFieldsConflict }); using (FileStream outFileStm = new FileStream(outFile, FileMode.Create)) { genDoc.Save(outFileStm); } } } } } // Example conflict resolution where we'll let the forms just bring in their fields normally, and only rename fields if needed // the rename here is VERY SIMPLE and may not be complex enough (we just pre-end "new" to the field name of any conflicting field" // You could do something involving random numbers/letters , GUID etc.. private static void ResolveFormFieldsConflict(object sender, FormFieldsConflictEventArgs args) { if (args.AreFieldTypesEqual) { // no conflict really so just let it ride args.ConflictResolution = FormFieldsConflictResult.KeepCurrentFieldAndMergeChildren; } else { // generate new name for field args.ExternalField.FieldName = "new" + args.ExternalField.FieldName; args.ConflictResolution = FormFieldsConflictResult.KeepBoth; } }
```

Original Article

Q10443 - INFO: Changes Introduced in DotImage 11.0

Last Update

INFO: Changes Introduced in DotImage 11.0

2026-11-06 - TD - (Fixed outdated links)

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50058/INFO-Changes-Introduced-in-DotImage-...