Changes Introduced != "release notes"

Please note, this document is intended to "go beyond" the basic release notes and give more details where needed. Such as Release notes may say "new feature - added single page view to WDV" where the Changes Introduced will tell you more useful information such as "to enable this feature you need to set the singlepage config option to true here..."

For basic Release notes, please see

INFO: 11.4 Full Release Notes

Known Breaking Issues

IMMEDIATE RELEASE

2023-Nov-15

v11.4.0.9 (Fix Pack 9) through v11.4.0.11 (Fix Pack 11) contain a regression that breaks DotTwain Device.GetSupportedResolutions()

This issue was resolved in Fix Pack 12 (11.4.0.12)

Please UPGRADE to 11.4.0.12 or later to resolve this issue

(original announcement November 1, 2023)

This non-exception generating issue will cause the result to always return 0 for the X and Y value of resolutions

Engineering is aware of this issue and is working on a fix. Customers using DotTwain and relying on this feature should use DotImage/DotTwain 11.4.0.8 until such a time as a fix is provided

BUG: DotTwain Device.GetSupportedResolutions() is returning TwainResolution with 0,0 for X and Y Values

INFO: Changes Introduced in DotImage 11.4

IMMEDIATE RELEASE:

2023-Nov-15

v11.4.0.11.410 (Fix Pack 11) contained a regression that broke "Assembly Tied License".

This issue has been addressed in v11.4.0.12.434 (Fix Pack 12).

Please UPGRADE to 11.4.0.12 or later to resolve this issue

(original announcement October 12, 2023)

This is a special type of license that most customers do not have / use, but if you have a "code based license" / "Cloud license" that is based on a "TiedAssembly" (not Strong name/public key tied), please do not use 11.4.0.11

Changes in Fix Packs

Some changes are introduced in Fix packs and Service packs after initial release. We note those here when they require changes/ add features that were not present

WDV: Improvements for Keyboard Interaction with Annotations

We've made a few improvements aimed at making it possible to better interact with annotations solely using a keyboard. This is intended as an accessibility and general usability improvement.

- tabindex field added to annotations - allows defining tab order for annotations per standard tabindex operation
- In support of TabIndex, we added changetabindex method annotation to change tabindex field value
- isAnnotationsPropertiesWindowShown method for annotations in order to be able to determine that annotation properties window is shown and tab (and other keys) should be ignored or processed in the specific way
- contextmenuPropertiesClicked event added

- showAnnotationsPropertiesWindow method to be able to open annotation properties window programmatically and potentially bind the action to the key
- annotationselected event added to annotationController
- annotationdeselected event added to annotationController

WDV Feature: Text Editor Key 'passthrough'

In 11.4.0.13 we added a feature in order to better enable non-mouse users to interact with WebDocumentViewer. It has always been possible to bind to keystrokes using standard JavaScript/jQuery in order to trigger any arbitrary code, including calling any API commands in our clientside API. However, while implementing a "keyboard interaction only" with annotations example, our support team noticed that once the text annotation text editor was open, the editor was "eating" all keydown events, such that there was no way to bind a key such as "escape" to call code to exit editing or similar

In order to address this, we added a new configuration option to WDV:

textAnnotationEditorIgnoredKeyCodes: [array of integer keycodes you want the editor to ignore/passthrough]

In order to use this, you need to set this in the main viewer AND if you are using the WebDocumentThumbnailer (WDT) also the WDT config

Below we'll give a simple example of passing the "ESC" key and thus being able to use it to close the editor (finishing editing)

```
ar _viewer = new Atalasoft.Controls.WebDocumentViewer({ // your usual config code here //
.... , textAnnotationEditorIgnoredKeyCodes: [27,9] }); var _thumbs = new
Atalasoft.Controls.WebDocumentThumbnailer({ // your usual config code here // .... ,
textAnnotationEditorIgnoredKeyCodes: [27,9] }); // example of binding to keydown event at the
global document level $(document.body).on("keydown", keydownHandler); // this will do the key
handling function keydownHandler(e) { switch (e.which) { case 27: //escape console.log("Esc
pressed!"); _viewer.annotations.hideEditor( curentTextAnnotationObjectHere ); break; case 9:
// tab key _viewer.annotations.hideEditor( curentTextAnnotationObjectHere ); break; default:
console.log("pressed: \n which: " + e.which + "\n key: " + e.key); break; } }
```

WDV Config Option: textannomobilezoom

starting in 11.4.0.12, we added a new WDV configuration option: textannomobilezoom

This feature defaults to true (the original behavior) but can not be disabled by setting to false

The setting must be made during initial configuration and can not be changed once set.

The issue is that on small screened mobile devices such as phones, the text editor size was considered so small as to be difficult to read/interact with, and so the default behavior of our editor was to zoom in, making text annotations quite a bit larger and thus easier to see. However, some customers (especially with the increase in adoption of larger screen tablets) find the zoom behavior highly undesirable

Thus, it can be disabled by setting

textannomobilezoom: false

in the WDV initialization/configuration

LibRaw Updaed

In 11.4.0.9, we updated RawDecoder to use LibRaw 0.21.

However there area actually two raw libs: DCRaw and LibRaw. The default RenderingEngine is DCRaw. At present, there is no direct property in RawDecoder to select the RenderingEngine, however it is possible to force. You need to add lines to your web.config or app.config.

If you just want to force LibRaw (recommended) you can use

```
Atalasoft> <RawDecoder RenderingEngine="LibRaw"/> </Atalasoft>
```

if you want to explicitly specify a libraw (let's say that a newer version addresses a new camera not in LibRaw-0.21

```
Atalasoft> <RawDecoder LibRawLocation="%PATH_TO_LIBRAW%\LibRaw-0.21.1\bin"
UserDefinedLibRaw="true" NativeLibValidationType="Signature" RenderingEngine="LibRaw"/>
```

```
</Atalasoft>
```

Tesseract 5 OCR Engine Added

As of 11.4.0.9, we introduced Tesseract5Engine to DotImage

Tesseract3Engine is still included (it may be removed in a future release), but for some time, Tesseract3.04 has been showing its age. Any new development should use Tesssaract5Engine instead of Tesseract3Engine, and all existing customers are strongly encouraged to move to Tesseract5Engine at their earliest convenience.

We are happy to announce we are now including Tesseract 5.3.0

To switch an application that used Tesseract3Engine to Tesseract5Engine:

1. Remove your reference to Atalasoft.dotImage.Ocr.Tesseract3.dll
2. Add a reference to Atalasoft.dotImage.Ocr.Tesseract5.dll
3. Replace any using or Imports statement explicitly referring to *Tesseract3* such as

   using Atalasoft.Ocr.Tesseract3; or Imports Atalasoft.Ocr.Tesseract3

   with

   using Atalasoft.Ocr.Tesseract5; or Imports Atalasoft.Ocr.Tesseract5
4. Change any explicit constructors or types in your code referring to *Tesseract3Engine* to *Tesseract5Engine*
5. For development/testing this should be all that is needed, but for deployment you will need to remove the old Tesseract 3.04 Tesseract OcrResources (the tessdata files) with the new ones which can be found in

   C:\Program Files (x86)\Atalasoft\DotImage 11.4\bin\OCRResources\Tesseract\v5.3.01

Tessdata Files For Additional Languages

Note that Tesseract 5 has many more languages than the 8 we ship with, you can download additional language files for Tesseract5Engine at

The Tesseract5 Training Data site

You would then download the desired files and drop those into your Tesseract5 OcrResources folder

C:\Program Files (x86)\Atalasoft\DotImage 11.4\bin\OCRResources\Tesseract\v5.3.0\

(Note that the 5.3.0 folder does NOT have a "*tessdata*" subfolder this is a change From Tesserac3Engine...)

Training Tesseract 5

If you can't find a suitable training file for your needs you can implement training yourself. Please note that Atalasoft does not provide support for training Tesseract OCR this link is provided for you as a convenience and is done so as-is

Please see The Official Tesseract Project Training Document

MAJOR Fix for Certificate Errors in WebCapture in Chrome

Changes in Google Chrome caused WingScan (WebCapture) service to become unreachable due to errors regarding certificates. This has been addressed in

11.4.0.4.0.828

A.K.A. 11.4 FixPack 4

If you are using WebCapture (WingScan) please immediately upgrade to 11.4.0.4 or newer to avoid this issue in Google Chrome on Windows.

HOWTO: Upgrade / Update DotImage - Including Web Apps Using WebDocumentViewer and/or WebCapture

A fix for Chrome on MacOS scanning is tentatively scheduled for FixPack 5 coming in early 2023

Download 11.4.0.4 on our official site: (May not be available until December - use Alternate

INFO: Changes Introduced in DotImage 11.4

Download Links below if needed)

ALTERNATE Download Link for 11.4.0.4 (available immediately)

SECONDARY ALTERNATE Download Link for 11.4.0.4 (available until official link is live)

NOTE: If you are using NuGet / NPM packages, these can lag as much as 2 weeks behind official publication due to the extended publishing process.

QRCodeOptions for Mirrored Codes

In FixPack 3 (11.4.0.3) we added a new property to QRCodeOptions to tell the engine to attempt to look for mirrored (horizontal) codes.

The reason for this is that some customers were seeing QR codes that look valid but would not read - the source was determined to be that the code was captured on a mobile device using a "back camera" which was using a mirror image.

Mirrored codes are technically not to the QR specification and so, not reading them is the "correct" result. However, mirrored images for QR codes are common due to QR codes ubiquity in situations where mobile devices cameras are the input device.. Enabling this new property will cause some performance hit as the memory and processing needed to do the Mirror operation, plus the processing needed for a second pass at detection. Thus, the default setting will be false and it will need to be explicitly set in your ReadOpts.QRCodeOptions if you want this automatic extra processing

The property name is DetectMirroredQRCodes

PDF option Now Added to DeliverablesConfig For WebCapture

"pdf" is now an additional supported option for DeliverablesConfig.format in WebCapture

Proper use is format: 'pdf'

NEW Features

.NET Core

In 11.4 we have begun shipping DLLs marked as "5.0" (C:\Program Files (x86)\Atalasoft\DotImage 11.4\bin\5.0)

These are DLLs for targeting .NET 5 / .NET 6 (Also known as .NET core) NOTE that we do not support .NET Core 3.1 or older with these dlls)

There will be more to come here, please check back in this space soon for links to articles and/or sample apps.

IMPORTANT NOTE about .NET Core and WinForms Designer

Unfortunately, our .NET 5 / .NET 6 components do not work with the Visual Studio WinForms forms designer. This is due to a known issue in Visual studio handling of .NET 5 and forms designer. We are currently investigating if /when the issue is addressed in .NET 6 and hope to be able to offer full support for WinForms designer mode when targeting .NET 6 in the future

For the time being, your options are

- Target .NET Framework instead of .NET 5 or .NET 6 - (This is your best option for full functionality)
- Target .NET Framework to do your designer work then up convert your project to target .NET 6 - (This is possible but the up convert is one-way (you can't back it down to .NET framework) meaning the designer will be set/not able to render with our controls - so any updates/fixes will have to then be done by adjusting the designer created code
- Manually handle the code for placing and configuring the controls in code - (This is possible but tedious - you'd need to essentially manually write al the code that the form designer would usually write on your behalf - it's not really a valid solution for any but the most experienced WinForms developers who don't mind not having a visual designer)

NON-BREAKING Changes

INFO: Changes Introduced in DotImage 11.4

These are important changes but ones that don't directly require code changes or other large rewrites

Documentation / Help Files

In previous versions, the ApiReference.chm and DevelopersGuide.pdf were installed to a folder named help to:

C:\Program Files (x86)\Atalasoft\DotImage 11.x\Help\

These files are now provided as download links in the DotImage 11.4 start menu folder

Developers Guide

START->All Programs->Atalasoft DotImage 11.4->DevelopersGuide

Direct Download

.NET API Reference

START->All Programs->Atalasoft DotImage 11.4->ApiReference

Direct Download

WebDocumentViewer (clientside) API Reference

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

WebCapture (Clientside) API Reference

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

BREAKING Changes

INFO: Changes Introduced in DotImage 11.4

These are larger changes that may require code changes/ rewrites or where features/support was dropped or removed

Web App / IIS licensing

NOTE: this issue should be addressed in 11.4.0.3 which has a tentative ETA of early October

A recent regression has come to light - Customers deploying .NET Framework web applications IIS may find that there's an issue with "Unable to find license file in the folllowing locations" even though they've placed their license files (correctly) in the app bin directory as before.

The root issue is that starting in 11.4.0.0, the licensing is looking in the temp cache files generated for the app. This is an issue under ongoing investigation, and we plan to resolve it in an upcoming fixpack.

WorkAround

The recommended workaround is to create a license directory on the server in question which the IIS process has rights to and use the BaseAssemblyPath Registry key value to point to this location per:

HOWTO: Configure your deployment to use Atalasoft license files in a custom location

Please pay special attention to the key path - there are two - one is for x64 (default IIS) HKLM\Software\Atalasoft\DotImage\{version}\ and the other is for 32 bit (uncommon) HKLM\Software\Wow6432Node\Atalasoft\DotImage\{version}\

DLL Restructuring

In 11.4 you may notice some DLLs are "missing" versus previous versions. This is by design.

We distribute our SDK in sets based on Framework

our dlls for targeting .NET Framework 3.5 are in

C:\Program Files (x86)\Atalasoft\DotImage 11.4\bin\3.5

Our DLLs for targeting .NET Framework 4.5.2 through 4.8.x are in

C:\Program Files (x86)\Atalasoft\DotImage 11.4\bin\4.5.2

Our DLLs for .NET 5 and .NET 6 are in

C:\Program Files (x86)\Atalasoft\DotImage 11.4\bin\5.0

for the 5.0 distributions of DLLs, we've removed

Atalasoft.DotAnnotate.Ink.dll

Atalasoft.dotImage.WebControls.Mvc.dll

These components may also be removed for the .NET Framework versions in future releases (support for Ink annotations and MVC legacy web controls are deprecated as are the Legacy web controls - new application development for web based projects should exclusively use our WebDocumentViewer (modern HTML5) web controls.

Dropped Support

MS Internet Explorer (11 and older), and non-chromium versions of MS Edge

We can no longer officially support MS IE 11 or older, and we can not support MS Edge if you're using a non-Chromium based version.

Your legacy apps MAY continue to work with these outdated browsers and if so that's OK but if you report a bug/issue with our web controls if must be replicible in Current versions of Google Chrome, MS Edge (Chromium based), and/or FireFox; or in the current Safari release for MACOS machines.

MACOS Mojave and older dropped

This applies to the Web Capture client-side component. The installable MACOS package that clients using pages with WebCapture need to install. The package no longer supports MACOS Mojave and older

INFO: Changes Introduced in DotImage 11.4

INK Annotation Support Removed

MS Ink annotation support is being removed. Its not part of the new .NET 5 dlls and may be removed from 3.5 and4.5.2 dlls in future. This was a seldom used feature as our Freehand annotations work very well to handle what they provided without needing a whole separate MS infrastructure

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50409/INFO-Changes-Introduced-in-DotImage-...