Release Notes - General

The full official release notes for all versions of DotImage can be found below on our official Release Notes Page

We've also combined all release notes for 11.2 into a single KB article available here: INFO: 11.2 Full Release Notes

This version is exceptionally useful as it's fully searchable via CTRL+F to quickly spot which fixpack a given fix or feature was released in.

Individual releases for 11.2

- Initial Release (December 2019) - v11.2.0.0
- Fix Pack 1 (January 27, 2020) - v11.2.0.1
- Fix Pack 2 (March 31, 2020) - v11.2.0.2
- Fix Pack 3 (May 4, 2020) - v11.2.0.3
- Fix Pack 4 (June 30, 2020) - v11.2.0.4
- Fix Pack 5 (July 15, 2020) - v11.2.0.5
- Fix Pack 6 (September 21, 2020) - v11.2.0.6
- Fix Pack 7 (November 2, 2020) - v11.2.0.7
- Fix Pack 8 (December 14, 2020) - v11.2.0.8
- Fix Pack 9 (January 28, 2021) - v11.2.0.9
- Fix Pack 10 (March 16, 2021) - v 11.2.0.10
- Fix Pack 11 (May 21, 2021) - v11.2.0.11
- Fix Pack 12 (June 28, 2021) - v11.2.0.12

Updates (needing code changes/explanation) Introduced in Fixpacks

v11.2.0.12.591

New Features

Added Optional Tiling (PDFs only) to WDV to Improve Apparent Performance

In fix pack 12, we added a new tiling feature to WebDocumentViewer which does not actually speed performance but gives faster visual feedback on image loading which makes the loading process "feel faster" to users.

This feature is disabled by default, and needs to be configured to be used

To use it you must add the tiling configuration option to your WebDocumentViewer config/initialization and note that if you are also using WebDocumentThumbnailer, you must also add the identical config there.

The option name is "tiling" and the valid values are false/null (default) or an integer representing the width/height of the square tile size (tiling adds some overhead and should be implemented only for "larger" images.. suggested minimum value would be 1000 pixels, but you could go as low as perhaps 256.

Example for enabling tiling at 1000px:

```
ar _viewer = new Atalasoft.Controls.WebDocumentViewer({ parent: $('#_containerViewer'),
toolbarparent: $('#_toolbar1'), serverurl: _serverUrl, allowannotations: true, savepath:
'Saved/', showscrollbars: true, forcepagefit: true, showselecttools: true, tiling: 1000 });
var _thumbs = new Atalasoft.Controls.WebDocumentThumbnailer({ parent: $('#_containerThumbs'),
serverurl: _serverUrl, documenturl: _docUrl, allowannotations: true, viewer: _viewer,
showscrollbars: true, tiling: 1000 });
```

NOTE: this only applies to images that are PDFs - other image types will not tile. Also, if you custom handle the ImageRequested event in your handler, Tiling will be ignored

v11.2.0.10.540 (FixPack 10)

New Features

Added Support for HEIF Image Format

In Fix Pack 10, we added support for Apple High Efficiency Image Format (HEIF). Note that this requires that you add a reference to our Atalasoft.dotImage.Heif.dll so that you can use the HeifDecoder and optionally add it to your RegisteredDecoders.Decoders collection.

Please see HOWTO: Add Support for HEIF Files to DotImage for mor

IMPORTANT: Support is for HEIF. The sister format HEIC is not supported. HEIC is a container format which could include video or burst shots. DotImage does not support Video or burst mode HEIC files - we support the HEIF format for still images.

Known Issues

API Reference documents incorrectly list the version as 11.3 instead of 10.2.0.10.

This is cosmetic only - the documentation is correct for 11.2.0.10. This will be resolved in the 11.2.0.11 fix pack. No action is required

v11.2.0.5.387 (FixPack 5)

New Features

A new feature was added to allow support for PdfDecoder to be used across app domains. This is a specific fix for issues when you have a single AppPool running multiple sites. Please do not use this unless you're having issues with PdfDecoder in this exact situation (it has possible performance hit and should only be used to solve a specific issue - if you're unsure, create a support case.

Updated source code to have configurable lock mechanism. In order to have to working with multiple app domains the following config section should be added to app.config (web.config):

```
<configSections> <section name="Atalasoft" type="Atalasoft.Shared.AtalasoftConfigSection,
Atalasoft.Shared"/> </configSections> <Atalasoft> <PdfDecoder
UseForMultipleAppDomains="true"/> </Atalasoft>
```

Note, that section should be the first section underneath xml element.

v11.2.0.4.372 (FixPack 4)

Breaking Changes

WebDocViewer resources - jQuery 3.5.1 now required

11.2.0.4.372 updated to use jQuery 3.5.1 as the minimum version. The updated jQuery file is included in:

```
:\Program Files (x86)\Atalasoft\DotImage 11.2\bin\WebResources\WebDocViewer
```

You will need to update your script inclusions to point to the new version

Updated links example

```
<script src="WebDocViewer/jquery-3.5.1.min.js" type="text/javascript"></script> <script
src="WebDocViewer/jquery-ui-1.12.1.min.js" type="text/javascript"></script> <script
src="WebDocViewer/raphael-min.js" type="text/javascript"></script> <script
src="WebDocViewer/clipboard.min.js" type="text/javascript"></script> <script
src="WebDocViewer/atalaWebDocumentViewer.js" type="text/javascript"></script> <link
href="WebDocViewer/jquery-ui-1.12.1.min.css" rel="Stylesheet" type="text/css" /> <link
href="WebDocViewer/atalaWebDocumentViewer.css" rel="Stylesheet" type="text/css" />
```

11.2.0.0 Initial Release

v11.2.0.0.224 - December 2019

Breaking Changes

Atalasoft developers work very hard to limit breaking changes in our SDK. We're happy to announce that we've avoided direct breaking chages requiring code changes in the use of our API. Breaking changes in this release are limited to necessary update to jQuery version, removal of Sliverlight components and removal of the SharePoint Web part.

Legacy Web Controls Require Local Files (Security)

Legacy WebAnnotationViewer, WebImageViewer, WebThumbnailViewer no longer open files from outside of the local web root. It is no longer permitted to open files by absolute path or any files not within virtual directories. This was done to comply with Veracode Flaw ID 480

Applications should copy needed files into a directory within the web root and if needed scavenge/clean out files older than 24 hours

Jp2Encoder Removed

In 11.1, we added in a new Jpeg2000Encoder and deprecated the Jp2Encoder with a warning. We have now removed the old Jp2Encoder from DotImage.. make sure you update to using Jpeg2000Encoder.

PdfDocument.Examiner.ExaminerResults

The ExaminerResults.FromStream(...) method has implemented a new overload:

```
ExaminerResults.FromStream(Stream stm,
                           System.Security.SecureString userPassword,
                           System.Security.SecureString ownerPassword)
```

The overload ExaminerResults.FromStream(Stream stm, string userPassword, string ownerPassword) has been Deprecated and will be removed in a future release.

Furthermore, There is now a compiler error due to an ambiguous call. If you had been passing in null values:

```
sing (FileStream fs = new FileStream(inFile, FileMode.Open)) { ExaminerResults res =
ExaminerResults.FromStream(fs, null, null); // ... }
```

You will need to change your code to either use the method that doesn't take password (if you do not need to pass in either user or owner pasword):

```
sing (FileStream fs = new FileStream(inFile, FileMode.Open)) { ExaminerResults res =
ExaminerResults.FromStream(fs); // ... }
```

Or, if you wish to supply passwords, do so using SecureString so that you will not be caught out when the constructor taking non secure string is removed:

```
sing (FileStream fs = new FileStream(inFile, FileMode.Open)) { string userPass =
"somePassword"; SecureString secureUserPass = new SecureString(); foreach (char ch in
userPass) { secureUserPass.AppendChar(ch); } string ownerPass = "someOtherPassword";
SecureString secureOwnerPass = new SecureString(); foreach (char ch in ownerPass) {
secureOwnerPass.AppendChar(ch); } ExaminerResults res = ExaminerResults.FromStream(fs,
secureUserPass, secureOwnerPass); // ... }
```

If you choose to call FromStream(strea, string, string) it will work, but you will receive a warning.. this will be removed in a future release.

WebDocumentViewer / WebCapture

11.2 has updated the minimum version of jQuery required from 3.3.1 (required in 11.1) to 3.4.1. jQueryUI has not changed.

This means you need to ensure when updating your solutions using WebDocumentViewer and/or WebCapture that you update the jQuery include call to point to jquery-3.4.1.min.js

old 11.1 definition

```
<script src="WebDocViewer/jquery-3.3.1.min.js" type="text/javascript"></script> <script
src="WebDocViewer/jquery-ui-1.12.1.min.js" type="text/javascript"></script> <script
src="WebDocViewer/raphael-min.js" type="text/javascript"></script> <script
src="WebDocViewer/clipboard.min.js" type="text/javascript"></script> <script
src="WebDocViewer/atalaWebDocumentViewer.js" type="text/javascript"></script> <link
href="WebDocViewer/jquery-ui-1.12.1.min.css" rel="Stylesheet" type="text/css" /> <link
href="WebDocViewer/atalaWebDocumentViewer.css" rel="Stylesheet" type="text/css" />
```

new 11.2 definition

```
<script src="WebDocViewer/jquery-3.5.1.min.js" type="text/javascript"></script> <script
src="WebDocViewer/jquery-ui-1.12.1.min.js" type="text/javascript"></script> <script
src="WebDocViewer/raphael-min.js" type="text/javascript"></script> <script
src="WebDocViewer/clipboard.min.js" type="text/javascript"></script> <script
src="WebDocViewer/atalaWebDocumentViewer.js" type="text/javascript"></script> <link
href="WebDocViewer/jquery-ui-1.12.1.min.css" rel="Stylesheet" type="text/css" /> <link
href="WebDocViewer/atalaWebDocumentViewer.css" rel="Stylesheet" type="text/css" />
```

Visual Studio Runtime Dependencies Change

Ever since Atalasoft added support for .NET 4x in our 9.0 release, we've had a hard dependency on the VS2010 C++ runtime Redistributable.

VS2010 has been retired for some time now, but we had (until 11.2) still relied on these outdated runtimes due to some third party dependencies.

DotImage 11.2 no longer depends on the VS2010 C++ Redistributable Runtime.. our .NET 4.5.2 dlls now depend on the VS2017 Visual Studio C++ runtime.

Please see: INFO: .NET 4.x Deployment Errors for more details.

Also please note that our 3.5 DLLs no longer require the VS2005 C++ Runtime Redistributable.. they require the VS2008 C++ Runtime Redistributable (.NET 3.5.1 support installed form Windows components)

SPECIAL NOTE: WingScan requires the VS 2015 C++ Runtime to be installed on client machines

ASP.NET Core Changes

11.2 supports ASP.NET Core 2.1 targeting the .NET framework 4.6.1 or above only.

Support for ASP.NET core 1.x and 2.0 have been dropped.

For more information, please see: FAQ: Support for ASP.NET Core / .NET Core

GlyphReader OCRResources

In 11.2, we upgraded to GlyphReader 5.0, so the OCRResources folder changed from:

C:\Program Files (x86)\Atalasoft\DotImage 11.1\bin\OCRResources\GlyphReader\v4.0

to

C:\Program Files (x86)\Atalasoft\DotImage 11.2\bin\OCRResources\GlyphReader\v5.0

INFO: Changes Introduced in DotImage 11.2

Please ensure you update the path info on any GlyphReaderLoader and update the GlyphReader OCRResources in your application to use the new 5.0 versions.

See also: the GlyphReader v5.0 section of

HOWTO: Deploy a project using OCR

Obsolete components

Support for the following has been removed in this version:

- **Jp2Encoder** - This class is removed - use Jpeg2000Encoder instead
- **SharePoint** - DotImage introduced a SharePoint WebPart in 10.4. This was a functional component, but due to the nature of WebPart architecture it was a bit of a "hood welded shut". It worked as is for the limited use case it was created for but could not be customized/improved. Our customers who used it generally agreed that it was "close, but without the ability to customize, it did not suit our needs". Adoption was so low that it was determined that our development resources would be better spent on other improvements/features. Users wishing to continue to use the SharePoint WebPart will need to stay with 11.1 or older.
  - Note that if you are using a WebDocumentViewer / WebCaputre solution hosted on IIS where you've back-ended to a SharePoint repository, we have not done anything to prevent that. This removal is only our specific DotImage SharePoint Web Part.
- **Silverlight** - Demand for this component was never high. MS has indicated they're killing the product line, and time and resources needed to update to Silverlight 5 were better used on other improvements/features. Users with Sliverlight 4 solutions will need to stay with DotImage 11.1 or older.

Updates / Improvements

Barcode recognition improvements

Recognition engine has been improved for the following symbologies:

- Code 39
- QR

- PDF417
- Code 128
- DataMatrix
- Aztec
- Patch Code

No code changes are required - Improvements are entirely internal

New Features

Modular Web Document Viewer

Module loaders support has been added to WDV so it can be used as an Asynchronous Module Definition (AMD) or CommonJS module. This feature avoids polluting the global namespace and reduces the risk of conflicting dependencies in the client

Our engineering team has created a document to explain how to set up WDV Tutorial: Setup Dependencies

PDF Security Improvements

PdfDocument, PdfGeneratedDocument, PdfDocumentSigner, PdfDocument.ExaminerResults have all had new constructor / method overloads added to utilize System.Security.SecureString as required for higher secure coding standards going forward.

IN the Breaking Changes section, we specifically mentioned PdfDocument.ExaminerResults as the old signature ExaminerResults.FromStream(Stream stm, string userPass, string ownerPass); is now ambiguous with a new method taking SecureString... this section covers other objects that do not give an error at this time but may require changes in future

PdfDocument SecureString Changes

We've deprecated several old constructors and methods to be removed in the future

Old / deprecated (to be removed in future)

```
ONSTRUCTORS PdfDocument(String userPassword, Stream stm) PdfDocument(String userPassword,
String filePath) PdfDocument(String userPassword, String ownerPassword, Stream)
PdfDocument(String userPassword, String ownerPassword, Stream stm, PdfDocumentLoadedProgress
prog) PdfDocument(String userPassword, String ownerPassword, String filePath, RepairOptions
opts) PdfDocument(String userPassword, String ownerPassword, Stream stm,
PdfDocumentLoadedProgress prog, RepairOptions opts) PdfDocument(String userPassword, String
ownerPassword, Stream stm, PdfDocumentLoadedProgress prog, RepairOptionsopts, Boolean
leaveOpen) PROPERTIES public string OwnerPassword { get; set; } public string UserPassword {
get; set; } METHODS Combine(String password, Stream outStream ,Stream[] inStreams)
Combine(String password, Stream outStream, String[] inFiles) Repair(String userPassword,
String ownerPassword, Stream inStream, Stream outStream, RepairOptions opts) Repair(String
userPassword, StringownerPassword, String inFilePath, String outFilePath, RepairOptions opts)
Save(String userPassword, Stream) Save(String userPassword, String) Save(String userPassword,
String ownerPassword, Stream) Save(String userPassword, String ownerPassword, String)
Separate(String userPassword, String ownerPassword, Stream inStream, PdfStreamCreator
creator) Separate(String userPassword, String ownerPassword, String inFilePath, String
destFolder, String fileNameFormat, Boolean overwrite)
```

New in 11.2

```
ONSTRUCTORS PdfDocument(Stream stm, RepairOptions opts) PdfDocument(Stream stm, SecureString
userPassword) PdfDocument(String filePath, RepairOptions opts) PdfDocument(String filePath,
SecureString userPassword) PdfDocument(Stream stm, SecureString userPassword, SecureString
ownerPassword) PdfDocument(String filePath, SecureString userPassword, SecureString
ownerPassword) PdfDocument(Stream stm, SecureString userPassword, SecureString ownerPassword,
PdfDocumentLoadedProgress prog) PdfDocument(String filePath, SecureString userPassword,
SecureString ownerPassword, RepairOptions opts) PdfDocument(Stream stm, SecureString
userPassword, SecureString ownerPassword, PdfDocumentLoadedProgress prog, RepairOptions opts)
PdfDocument(Stream stm, SecureString userPassword, SecureString ownerPassword,
PdfDocumentLoadedProgress prog, RepairOptions, Boolean) METHODS Repair(Stream inStm, Stream
outStm, RepairOptions opts, SecureString userPassword, SecureString ownerPassword)
Separate(Stream inStm, PdfStreamCreator creator, SecureString userPassword, SecureString
ownerPassword) Separate(String inFilePath, String outDirectory, String fillenameFormat,
Boolean overwrite, SecureString userPassword, SecureString ownerPassword)
```

SecureString changes: PdfGeneratedDocument, PdfDocument, PdfDocumentSigner, PdfDocument.ExaminerResults

All classes based on PdfDocument (PdfDocument, PdfGeneratedDocument, PdfDocumentSigner, PdfDocument.ExaminerResults) are deprecating password properties from simple string values. The specification for PDF wants these to come from SecureString. To this end, we've now added new constructors and properties that take SecureString instead of simple string for those fields. For the time being, the old plain text string versions remain usable, but will cause warnings/notices in Visual Studio for deprecation.

Support strongly encourages customers to switch to the new constructors, method calls and properties that use SecureString at your earliest convenience so that when the old properties, methods, and constructors are removed, you will not be caught with an emergency breaking change.

There is no official ETA on when we will remove those items, so please begin the migration now.

Supported Environments

Supported Web Browsers

- Google Chrome
- Mozilla Firefox
- MS Internet Explorer 11 or above
- Microsoft Edge
- Apple Safari (supported on macOS only)

Note: Only vendor supported versions of a browser will be supported

MAC OS Scanning

- OSX Sierra (10.12) or above (*)
- Requires use of a scanner that is supported by Apple's Image Capture Application. see apple's list of supported scanning devices.

*** NOTE:** There is currently a known issue where MAC OS Catalina (10.15) does not work. This is expected to be addressed in a future fixpack.

Supported Mobile Browsers

Mobile browser support relates to support for viewing from a mobile device of a site built using our modern WebDocumentViewer / WebDocumentThumbnailer controls hosted on IIS in an ASP.NET or ASP.NET Core web application.

- Google Chrome (only on Android)
- Apple Safari (Only on iOS devices)

Note: Only vendor supported versions of a browser will be supported

Windows Operating Systems

Client Operating Systems

- Windows 10
- Windows 8.1 (desktop mode only)
- Windows 7.1 SP1

Server Operating Systems

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 SP1

Note: We only officially support OS versions which are currently within their supported by Microsoft life cycle

Remote Desktop Technologies

The following remote desktop technologies are supported:

- Citrix Virtual Apps (formerly XenApp)
- Citrix Virtual Desktops (formerly XenDesktop)
- Citrix XenApp And XenDesktop (> 6.5)
- Windows Remote Desktop Services for supported Windows versions

Dropped in this release:

- Citrix XenApp 6.5

- Citrix XenDesktop 6.5

Citrix support for scanning requires the Citrix TWAIN Redirection to be enabled.

Note: Only versions (include LTSR) of Citrix Virtual Apps/Desktops and XenApp/XenDesktop supported by Citrix are supported by our SDK. See Citrix product matrix page for "Citrix Virtual Apps and Desktops" and "XenApp & XenDesktop" for more information.

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50315/INFO-Changes-Introduced-in-DotImage-...