

INFO: HEIC / HEIF Support in DotImage

11.2.0.10 added support for High Efficiency Image Format (HEIF) images

A word about format support

Due to the way the file extension is used - our decoder is the HeifDecoder, but you may find that files being processed have the HEIC extension

HEIF is a container format developed by the Moving Picture Experts Group (MPEG) which can contain many different licensed audio and video formats

Atalasoft supports a subset of these images - most often seen as output from Apple devices as it has been Apples go to format for some time now.

Our HeifDecoder supports single and multiple frame HEIC / HEIF images

See also: [What the HEIC? Apple's Highly Compressed Image Format Still Confuses](#)

Adding HEIF/HEIC support to DotImage

This topic is covered by its own KB article: [HOWTO: Add Support for HEIF \ HEIC Files to DotImage](#)

Multiple Frames in HEIF / HEIC Images

The HEIF / HEIC formats can support multiple frames - such as from a burst mode (however, please note that in testing, Apple devices save such files as .MOV (mpeg 4 movie) files and not HEIF/ HEIC and thus these files can not be decoded by DotImage. We support HEIF files with multiple HEIC frames

To make use of multiple frames, you need to use the GetFrameCount() method of the HeifDecoder

```
int imagesCount = heifDecoder.GetFrameCount(stream); for (int i = 0; i < imagesCount; i++) {
using (AtalaImage atalaImage = heifDecoder.Read(stream, i, null)) { // .. process image here
} }
```

Making use of features Unique to HeifDecoder

There are a couple features unique to HeifDecoder

In order to get access to auxiliary images, thumbnails, metadata, etc.

GetHeifDocument(Stream, ProgressEventHandler) method should be used:

```
HeifDocument heifDocument = heifDecoder.GetHeifDocument(stream, null); foreach (var heifImage in heifDocument.HeifImages) { // reading auxiliary images foreach (var auxiliaryImage in heifImage.AuxiliaryImages) { // process auxiliary image } // retrieve AtalaImage var atalaImage = heifImage.GetImage(); }
```

Configuring for External Library (Bring your own lib)

Valid values for NativeLibValidationType are CheckSum, Signature, and None. NOTE that Signature does not work in all instances due to how some lib versions are not signed. Use CheckSum or None for best compatibility.

HeifLibLocation can be a relative or full path. Take care with paths in web context that the web server has access.

Configuring in web.config

For a web app, you would configure the RawDecoder in web.config

```
configSections> <section name="Atalasoftware" type="Atalasoftware.Shared.AtalasoftwareConfigSection, Atalasoftware.Shared"/> </configSections> <Atalasoftware> <HeifDecoder HeifLibLocation="." NativeLibValidationType="Signature"/> </Atalasoftware> <connectionStrings />
```

Configuring in App.Config

For desktop apps in .NET Framework, you configure in app.config

```
?xml version="1.0" encoding="utf-8"?> <configuration> <configSections> <section name="Atalasoftware" type="Atalasoftware.Shared.AtalasoftwareConfigSection, Atalasoftware.Shared"/> </configSections> <Atalasoftware> <HeifDecoder HeifLibLocation="." NativeLibValidationType="Signature"/> </Atalasoftware> </configuration>
```

INFO: HEIC / HEIF Support in DotImage

Atalasoft Knowledge Base

<https://www.atalasoft.com/kb2/KB/50386/INFO-HEIC-HEIF-Support-in-DotImage>