

METADATA INSPECTOR

Metadata Inspector is a plugin that allows for the viewing on metadata within the Unity Inspector as well as at runtime. Version 1.0 supports reading metadata. Writing or saving metadata is not currently possible. Metadata Inspector is a wrapper for the Metadata Extractor project:

<https://drewnoakes.com/code/exif/>

Supported Image file types:

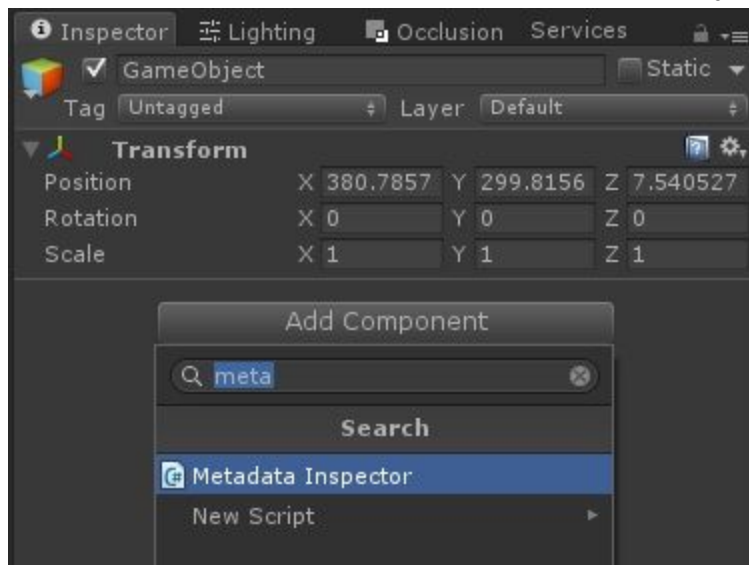
JPEG, PNG, WebP, GIF, ICO, BMP, TIFF, PSD, PCX, RAW, CRW, CR2, NEF, ORF, RAF, RW2, RWL, SRW, ARW, DNG, X3F.

Supported Metadata formats:

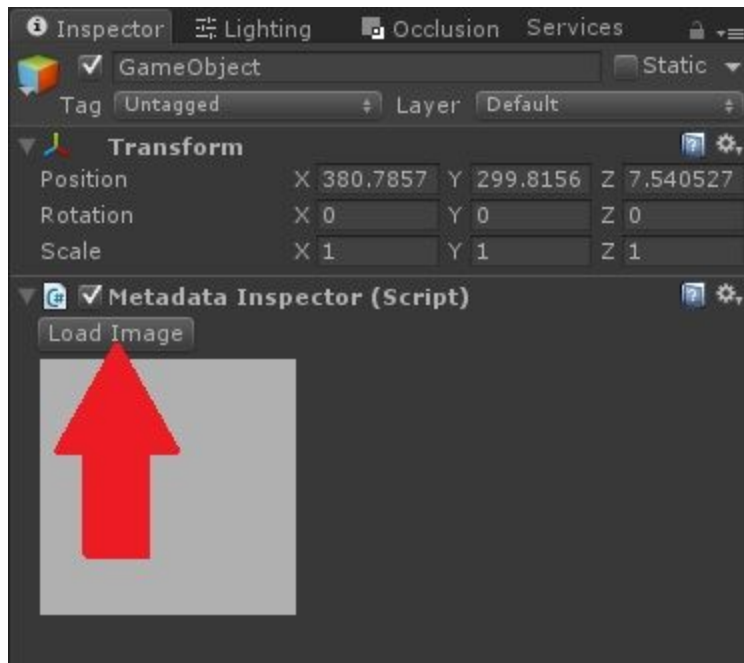
Exif, IPTC, XMP, JFIF, JFXX, ICC, 8BIM

How To Use in Editor:

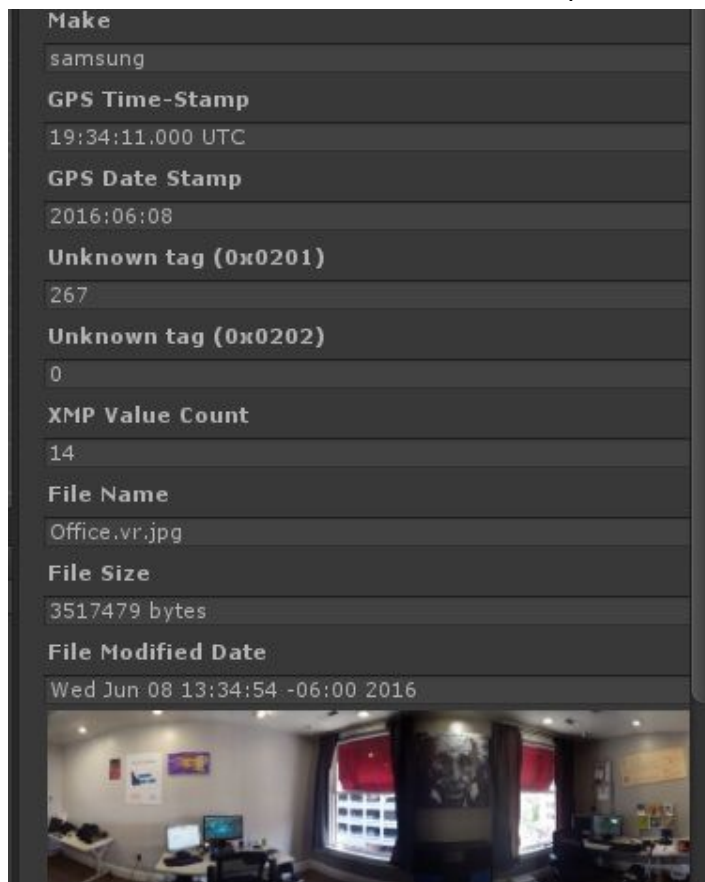
1. Add a “Metadata Inspector” script to a GameObject:



2. Click “Load Image” and select an image on the disk.



3. You can now view all Metadata in the Inspector:



How To Use At Runtime:

Define a variable for the MetadataInspector script:

```
// The MetadataInspector script  
public MetadataInspector myMetadataInspector;
```

Now use ***myMetadataInspector.LoadMetadataFromPath(string)*** to load a local file:

```
// Get the path from the GUI box  
myMetadataInspector.LoadMetadataFromPath(localPath);
```

And use ***myMetadataInspector.LoadMetadataFromURL(string)*** to load from a URL:

```
// Get the URL from the GUI box  
myMetadataInspector.LoadMetadataFromURL(imageURL);
```

Now that Metadata has been loaded, a Dictionary has been created with **(Key, Value)** corresponding to the Metadata. More information about Dictionaries can be found here:

<https://unity3d.com/learn/tutorials/modules/intermediate/scripting/lists-and-dictionaries>

<https://msdn.microsoft.com/en-us/library/xfhwa508.aspx>

The Dictionary is defined at: ***myMetadataInspector.metadata*** so to iterate through all of the metadata values, you can use:

```
// For each of the metadata keys  
foreach (string key in myMetadataInspector.metadata.Keys) {  
    // Log the Dictionary Keys and Values  
    Debug.Log(key + ": " + myMetadataInspector.metadata[key]);  
}
```

How To Find Specific Metadata:

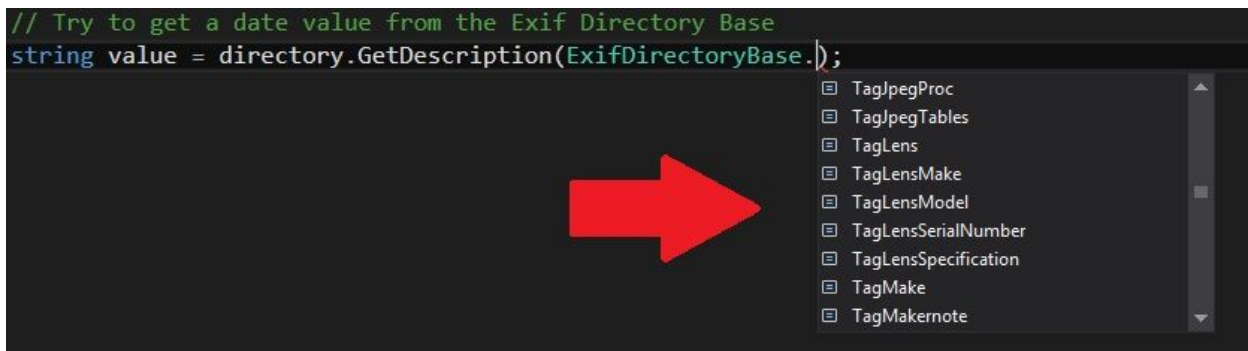
You can search through the dictionary for known text you want to find:

```
// Try to find a date value
private void GetDateMethodA() {
    // For each key in the metadata dictionary
    foreach (string key in myMetadataInspector.metadata.Keys) {
        // if key has "date or Date"
        if (key.Contains("date") || key.Contains("Date")) {
            // Update the GUI text
            textDateTime.text = myMetadataInspector.metadata[key];
        }
    }
}
```

Or you can try searching by using Metadata Extractor's extensive directory of Tags:

```
// Try to find a Compression value
private void GetCompressionMethodB() {
    // For each of the directories
    foreach (Directory directory in myMetadataInspector.directories) {
        // Try to get a date value from the Exif Directory Base
        string value = directory.GetDescription(ExifDirectoryBase.TagCompression);
        // If a value was found
        if (value != null) {
            // Update the GUI text
            textCompression.text = value;
        }
    }
}
```

In the example above, the script attempts to find a "Compression" value in the Metadata, by searching with ***Exif.DirectoryBase.TagCompression*** . There is a huge list of values you can search for, here is an example:



How to get additional support:

This is one of my early attempts at creating a tool for other developers and I strive to provide the best support possible. I will respond quickly to email send to:

nurfacegames@gmail.com

Video Intro / tutorial for this asset:

<https://www.youtube.com/watch?v=O4iNQ-r-FBY>

Info about Dictionaries in Unity:

This project saves metadata in dictionaries, you can find more information about dictionaries here:

<https://unity3d.com/learn/tutorials/modules/intermediate/scripting/lists-and-dictionaries>

<https://msdn.microsoft.com/en-us/library/xfhwa508.aspx>

Metadata Extractor Project:

For more information about the Metadata Extractor project please see the website here:

<https://drewnoakes.com/code/exif/>