

# Correct Image Orientation in FileMaker

Sometimes you may see pictures in FileMaker with wrong orientation. But in your image editing application, the orientation is correct. So you may wonder what is going on?



Image files like JPEG, PNG and TIFF have all metadata which describes the orientation of the image in 8 different ways. So the pixels in the picture come in the orientation they are read from the sensor chip. But you may hold your camera rotated, so the pixels flow into the file and the camera marks the image to be taken rotated and writes that in metadata.

Sadly not all applications handle this and show the image in the right orientation. For all 8 ways, we change the orientation to always be in mode 1 and save the image with the new orientation, so all applications will show the image correctly:

```
# load image
Set Variable [ $Image ; Value: MBS( "GMImage.NewFromContainer"; Correct Image
Orientation::Input ) ]
If [ MBS("IsError") = 0 ]
    #
    Set Variable [ $orientation ; Value: MBS( "GMImage.GetOrientation"; $Image ) ]
    If [ $orientation > 0 ]
        If [ $orientation = 1 ]
            # TopLeftOrientation = 1, (Line direction: Left to right, Frame Direction:
Top to bottom)
            # nothing to do
        Else If [ $orientation = 2 ]
            # TopRightOrientation = 2, (Line direction: Right to left, Frame
Direction: Top to bottom)
            # Flip horizontally
            Set Variable [ $r ; Value: MBS( "GMImage.Flop"; $Image ) ]
        Else If [ $orientation = 3 ]
            # BottomRightOrientation = 3, (Line direction: Right to left, Frame
Direction: Bottom to top)
            # Rotate 180°
            Set Variable [ $r ; Value: MBS( "GMImage.Rotate"; $Image; 180 ) ]
        Else If [ $orientation = 4 ]
            # BottomLeftOrientation = 4, (Line direction: Left to right, Frame
Direction: Bottom to top)
            # Flip vertically
```

```

        Set Variable [ $r ; Value: MBS( "GMImage.Flip"; $Image ) ]
    Else If [ $Orientation = 5 ]
        # LeftTopOrientation = 5, (Line direction: Top to bottom, Frame
Direction: Left to right)
        # Rotate 90° and flip horizontally
        Set Variable [ $r ; Value: MBS( "GMImage.Rotate"; $Image; 90 ) ]
        Set Variable [ $r ; Value: MBS( "GMImage.Flop"; $Image ) ]
    Else If [ $Orientation = 6 ]
        # RightTopOrientation = 6, (Line direction: Top to bottom, Frame
Direction: Right to left)
        # Rotate 90°
        Set Variable [ $r ; Value: MBS( "GMImage.Rotate"; $Image; 90 ) ]
    Else If [ $Orientation = 7 ]
        # RightBottomOrientation = 7, (Line direction: Bottom to top, Frame
Direction: Right to left)
        # Rotate -90° and flip horizontally
        Set Variable [ $r ; Value: MBS( "GMImage.Rotate"; $Image; -90 ) ]
        Set Variable [ $r ; Value: MBS( "GMImage.Flop"; $Image ) ]
    Else If [ $Orientation = 8 ]
        # LeftBottomOrientation = 8 (Line direction: Bottom to top, Frame
Direction: Left to right)
        # Rotate -90°
        Set Variable [ $r ; Value: MBS( "GMImage.Rotate"; $Image; -90 ) ]
    End If
    #
    Set Variable [ $r ; Value: MBS( "GMImage.SetOrientation"; $Image; 1 ) ]
    Set Variable [ $r ; Value: MBS( "GMImage.SetEXIFOrientation"; $Image;
1 ) ]
End If
#
# Write back to new field
Set Field [ Correct Image Orientation::Output ;
MBS( "GMImage.WriteToJPEGContainer"; $Image; GetAsText(Correct Image
Orientation::Input) ) ]
#
# free memory
Set Variable [ $r ; Value: MBS( "GMImage.Destroy"; $Image ) ]
End If

```

The example database will be included in future plugin releases.