

[Accelerate](#) / vImageConvert\_YpCbCrToARGB\_GenerateConversion(\_:\_:\_:\_:\_:\_:\_:\_)

## Function

# vImageConvert\_YpCbCrToARGB \_GenerateConversion(\_:\_:\_:\_:\_:\_:\_:\_:\_)

Generates the information that describes the conversion from YpCbCr to ARGB.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 13.1+ | macOS 10.10+ | tvOS 8.0+ | visionOS 1.0+ | watchOS 1.0+

```
func vImageConvert_YpCbCrToARGB_GenerateConversion(  
    _ matrix: UnsafePointer<vImage_YpCbCrToARGBMatrix>,  
    _ pixelRange: UnsafePointer<vImage_YpCbCrPixelRange>,  
    _ outInfo: UnsafeMutablePointer<vImage_YpCbCrToARGB>,  
    _ inYpCbCrType: vImageYpCbCrType,  
    _ outARGBTType: vImageARGBTType,  
    _ flags: vImage_Flags  
) -> vImage_Error
```

## Parameters

### matrix

A pointer to [vImage\\_YpCbCrToARGBMatrix](#) that contains the matrix coefficients for the conversion.

### pixelRange

A pointer to [vImage\\_YpCbCrPixelRange](#) that contains the pixel range information for the conversion.

### outInfo

A pointer to [vImage\\_YpCbCrToARGB](#) that's initialized with information for the conversion function to use later.

### inYpCbCrType

A [vImageYpCbCrType](#) to specify the input (YpCbCr) format.

### outARGBType

A [vImageARGBType](#) to specify the output (ARGB) format.

### flags

The options to use when performing this operation. Set the [kvImagePrintDiagnosticsToConsole](#) flag to print debug messages when a problem occurs.

## Return Value

[kvImageNoError](#); otherwise, one of the error codes in [Data Types and Constants](#).

## Discussion

You use this function to create the [vImage\\_YpCbCrToARGB](#) conversion information necessary for all YUV-to-RGB conversion functions.

The following example shows how to prepare for the conversion of a YUV format with ITU 601 video range to ARGB8888:

```
vImage_Error err = kvImageNoError;
vImage_Flags flags = kvImageNoFlags;
vImage_YpCbCrPixelRange pixelRange;
vImage_YpCbCrToARGB outInfo;

pixelRange.Yp_bias      = 16;           // The encoding for Y' = 0.0.
pixelRange.CbCr_bias    = 128;          // The encoding for CbCr = 0.0.
pixelRange.YpRangeMax   = 235;          // The encoding for Y'= 1.0.
pixelRange.CbCrRangeMax = 240;          // The encoding for CbCr = 0.5.
pixelRange.YpMax         = 255;          // A clamping limit above which the value is
pixelRange.YpMin         = 0;            // A clamping limit below which the value is
pixelRange.CbCrMax       = 255;          // A clamping limit above which the value is
pixelRange.CbCrMin       = 0;            // A clamping limit above which the value is
                                         //                                     ( pixelRange.CbCr_bias - (
```

  

```
err = vImageConvert_YpCbCrToARGB_GenerateConversion(kvImageITU601_YpCbCrToARGBMatrix);
```

The following example shows how you might define your own conversion coefficients:

```

vImage_YpCbCrToARGBMatrix matrix;
vImage_YpCbCrPixelRange pixelRange;

matrix.Yp          = 1.0f;
matrix.Cb_G       = -0.3441f;
matrix.Cb_B       = 1.772f;
matrix.Cr_R       = 1.402f;
matrix.Cr_G       = -0.7141f;
pixelRange.Yp_bias      = 16;           // The encoding for Y' = 0.0.
pixelRange.CbCr_bias    = 128;          // The encoding for CbCr = 0.0.
pixelRange.YpRangeMax   = 235;          // The encoding for Y'= 1.0.
pixelRange.CbCrRangeMax = 240;          // The encoding for CbCr = 0.5.
pixelRange.YpMax        = 255;          // A clamping limit above which the value is
pixelRange.YpMin        = 0;             // A clamping limit below which the value is
pixelRange.CbCrMax      = 255;          // A clamping limit above which the value is
pixelRange.CbCrMin      = 0;             // A clamping limit above which the value is
                                         // ( pixelRange.CbCr_bias - 1
err = vImageConvert_YpCbCrToARGB_GenerateConversion(&matrix, &pixelRange, &outInfo,

```

The `vImage_YpCbCrToARGB` structure this function creates can be reused concurrently, multiple times from multiple threads.

The conversions that are available are:

	RGB8	RGB16Q12	RGB16
YUV8	Y	N	N
YUV10	Y	Y	N
YUV12	Y	Y	N
YUV14	Y	N	Y
YUV16	Y	N	Y

## See Also

### Generating conversion information

`struct vImageYpCbCrType`

Constants that describe the encoding of a YpCbCr image for conversions between RGB and YpCbCr.

`struct vImageARGBType`

Constants that describe the encoding of an ARGB image for conversions between RGB and YpCbCr.

`struct vImage_YpCbCrToARGBMatrix`

The 3 x 3 matrix that the vImage library uses to convert from YpCbCr to RGB.

`struct vImage_YpCbCrToARGB`

The information that describes the conversion from YpCbCr to ARGB.

`struct vImage_YpCbCrPixelRange`

The description of range and clamping information for YpCbCr pixel formats.