

# pnoJpegLib 2 Documentation

## ***pnoJpeg2Create***

<b>Purpose</b>	Create a pnoJpeg2-Data-Structure
<b>Prototype</b>	<b>Err pnoJpeg2Create(UInt16 refNum, pnoJpeg2Ptr *data)</b>
<b>Parameters</b>	<- data            Pointer to the new Data
<b>Result</b>	errNone memErrInvalidParam memErrNotEnoughSpace
<b>Comments</b>	You have to call pnoJpeg2Free after reading the image to free the data
<b>See also</b>	pnoJpeg2Free

## ***pnoJpeg2LoadFromPtr***

<b>Purpose</b>	Load the JPEG-Data from a Pointer
<b>Prototype</b>	<b>Err pnoJpeg2LoadFromPtr(UInt16 refNum, pnoJpeg2Ptr data, MemPtr dataPtr, UInt32 dataSize)</b>
<b>Parameters</b>	-> data            Pointer to a pnoJpeg2-Structure -> dataPtr        Pointer to the JPEG-Data -> dataSize        size of the Data
<b>Result</b>	errNone memErrInvalidParam

## ***pnoJpeg2LoadFromVFS***

<b>Purpose</b>	Load the JPEG-Data from a VFS-File
<b>Prototype</b>	<b>Err pnoJpeg2LoadFromVFS(UInt16 refNum, pnoJpeg2Ptr data, UInt16 volRefNum, char *filePath)</b>
<b>Parameters</b>	-> data            Pointer to a pnoJpeg2-Structure -> volRefNum        number of the volume -> filePath         path to the file
<b>Result</b>	errNone memErrInvalidParam

## ***pnoJpeg2LoadFromHandle***

**Purpose** Load the JPEG-Data from a MemHandle

**Prototype** `Err pnoJpeg2LoadFromHandle(UInt16 refNum, pnoJpeg2Ptr data, MemHandle hImageData)`

**Parameters**

- > data Pointer to a pnoJpeg2-Structure
- > hImageData a Handle that contains the JPEG-Data

**Result**

- errNone
- memErrInvalidParam

## ***pnoJpeg2LoadFromFileStream***

**Purpose** Load the JPEG-Data from a FileStream

**Prototype** `Err pnoJpeg2LoadFromFileStream(UInt16 refNum, pnoJpeg2Ptr data, FileHand fh)`

**Parameters**

- > data Pointer to a pnoJpeg2-Structure
- > fh a filehandle that contains the JPEG-Data

**Result**

- errNone
- memErrInvalidParam

**Comments** Thanks to Jean-Pierre Morfin for this function

## ***pnoJpeg2GetInfo***

**Purpose** Read the Image Dimensions

**Prototype** `Err pnoJpeg2GetInfo(UInt16 refNum, pnoJpeg2Ptr data, Coord *width, Coord *height)`

**Parameters**

- > data Pointer to a pnoJpeg2-Structure
- <- width width of the image
- <- height height of the image

**Result**

- errNone
- memErrInvalidParam

**Comments** after getting the info about the Dimensions it's possible to change some loading settings, such as maxWidth, gayscale or scalefactor

### ***pnoJpeg2SetScaleFactor***

<b>Purpose</b>	Sets the Scalefactor for the image loading
<b>Prototype</b>	<code>Err pnoJpeg2SetScaleFactor(UInt16 refNum, pnoJpeg2Ptr data, UInt16 factor)</code>
<b>Parameters</b>	-> data                      Pointer to a pnoJpeg2-Structure -> factor                      down-scale-factor
<b>Result</b>	errNone memErrInvalidParam
<b>Comments</b>	possible factors: 1, 2, 4, 8

### ***pnoJpeg2SetGrayscale***

<b>Purpose</b>	Sets the Scalefactor for the image loading
<b>Prototype</b>	<code>Err pnoJpeg2SetGrayscale(UInt16 refNum, pnoJpeg2Ptr data, Boolean grayscale)</code>
<b>Parameters</b>	-> data                      Pointer to a pnoJpeg2-Structure -> grayscale                      true: load grayscale image; false: load color Image
<b>Result</b>	errNone memErrInvalidParam

### ***pnoJpeg2SetMaxDimensions***

<b>Purpose</b>	Set the maximum Dimensions for the image loading
<b>Prototype</b>	<code>Err pnoJpeg2SetMaxDimensions(UInt16 refNum, pnoJpeg2Ptr data, Coord maxWidth, Coord maxHeight)</code>
<b>Parameters</b>	-> data                      Pointer to a pnoJpeg2-Structure -> maxWidth ->maxHeight
<b>Result</b>	errNone memErrInvalidParam
<b>Comments</b>	Thanks to Jean-Pierre Morfin for this function

### ***pnoJpeg2Read***

**Purpose** Reads the data and decode it to the Bitmap

**Prototype** `Err pnoJpeg2Read(UInt16 refNum, pnoJpeg2Ptr data, BitmapPtr *bmpPtr)`

**Parameters** `-> data` Pointer to a pnoJpeg2-Structure  
`<- bmpPtr` Contains the loaded Image as Bitmap

**Result** `errNone`  
`memErrInvalidParam`

### ***pnoJpeg2Free***

**Purpose** Frees the pnoJpeg2 data structure

**Prototype** `Err pnoJpeg2Free(UInt16 refNum, pnoJpeg2Ptr *data)`

**Parameters** `-> data` Pointer to the Pointer of a pnoJpeg2-Structure

**Result** `errNone`  
`memErrInvalidParam`

### ***pnoJpeg2Version***

**Purpose** Returns the Version of the installed lib

**Prototype** `UInt16 pnoJpeg2Version(UInt16 refNum)`

**Parameters** `none`

**Result** Returns the Version of the installed Lib

## ***pnoJpeg2Resize***

<b>Purpose</b>	Does a fast resize with the new Dimensions
<b>Prototype</b>	<code>BitmapPtr pnoJpeg2Resize(UInt16 refNum, BitmapPtr inBmp, Coord newWidth, Coord newHeight, Err *error)</code>
<b>Parameters</b>	-> inBmp: the source Bitmap (it must be a 16-Bit-Bitmap) -> newWidth: -> newHeight: <- error: is set to the error code if an error happens
<b>Result</b>	Pointer to the new created Bitmap NULL if an error happens

## ***pnoJpeg2Resample***

<b>Purpose</b>	Does an antialiased resize with the new Dimensions
<b>Prototype</b>	<code>BitmapPtr pnoJpeg2Resample(UInt16 refNum, BitmapPtr inBmp, Coord newWidth, Coord newHeight, Err *error)</code>
<b>Parameters</b>	-> inBmp: the source Bitmap (it must be a 16-Bit-Bitmap) -> newWidth: -> newHeight: <- error: is set to the error code if an error happens
<b>Result</b>	Pointer to the new created Bitmap NULL if an error happens

## ***pnoJpeg2Bmp2DoubleDensity***

<b>Purpose</b>	Create a BitmapPtrV3 from the given Bitmap
<b>Prototype</b>	<code>Err pnoJpeg2Bmp2DoubleDensity(UInt16 refNum, BitmapPtr inBmp, BitmapPtrV3 *outBmp)</code>
<b>Parameters</b>	->inBmp: source ->outBmp: Pointer to the destination Bitmap
<b>Result</b>	errNone

## ***pnoJpeg2CreateThumbnail***

<b>Purpose</b>	Creates a new Bitmap wich is proportional resized
<b>Prototype</b>	<code>BitmapPtr pnoJpeg2CreateThumbnail(UInt16 refNum, BitmapPtr bmp, Coord maxWidth, Coord maxHeight, int method)</code>
<b>Parameters</b>	-> bmpmp: the source Bitmap (it must be a 16-Bit-Bitmap) -> maxWidth: -> maxHeight: ->method: pnoResize or pnoResample
<b>Result</b>	Pointer to the new created Bitmap NULL if an error happens

pnoJpegLib 2 Documentation.....	1
pnoJpeg2Create.....	1
pnoJpeg2LoadFromPtr.....	1
pnoJpeg2LoadFromVFS.....	1
pnoJpeg2LoadFromHandle.....	2
pnoJpeg2LoadFromFileStream.....	2
pnoJpeg2GetInfo.....	2
pnoJpeg2SetScaleFactor.....	3
pnoJpeg2SetGrayscale.....	3
pnoJpeg2SetMaxDimensions.....	3
pnoJpeg2Read.....	4
pnoJpeg2Free.....	4
pnoJpeg2Version.....	4
pnoJpeg2Resize.....	5
pnoJpeg2Resample.....	5
pnoJpeg2Bmp2DoubleDensity.....	5
pnoJpeg2CreateThumbnail.....	6