

# **Release Notes for Image Processing Toolbox™**

---

## How to Contact MathWorks



[www.mathworks.com](http://www.mathworks.com) Web  
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab) Newsgroup  
[www.mathworks.com/contact\\_TS.html](http://www.mathworks.com/contact_TS.html) Technical Support



[suggest@mathworks.com](mailto:suggest@mathworks.com) Product enhancement suggestions  
[bugs@mathworks.com](mailto:bugs@mathworks.com) Bug reports  
[doc@mathworks.com](mailto:doc@mathworks.com) Documentation error reports  
[service@mathworks.com](mailto:service@mathworks.com) Order status, license renewals, passcodes  
[info@mathworks.com](mailto:info@mathworks.com) Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.  
3 Apple Hill Drive  
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

*Release Notes for Image Processing Toolbox™*

© COPYRIGHT 2000–2012 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

### Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

### Patents

MathWorks products are protected by one or more U.S. patents. Please see [www.mathworks.com/patents](http://www.mathworks.com/patents) for more information.

## R2012b

Image gradient computation with <code>imgradient</code> and <code>imgradientxy</code> functions .....	2
Histogram matching with <code>imhistmatch</code> function .....	3
Multilevel thresholding with <code>multithresh</code> and <code>imquantize</code> functions .....	4
3-D image registration with <code>imregister</code> function .....	5
Code generation for <code>bwmorph</code> and <code>bwlookup</code> with MATLAB Coder .....	6
Added function <code>bwlookup</code> .....	7
Writing private metadata when anonymizing DICOM files .....	8
Expanded color options with <code>imshowpair</code> .....	9
Performance improvements .....	10
New Example .....	11

## R2012a

Intensity-Based Image Registration .....	14
Two New Functions to Visually Compare Images .....	15
Circle Detection Using the Circular Hough Transform ...	16
Performance Improvements .....	17
New and Updated Demos .....	18
Data Type Change to Output Variable for <code>bwdist</code> .....	19
<code>iradon</code> Function Updated .....	20
Functions Being Removed .....	21

## R2011b

Parallel Block Processing Now Possible with <code>blockproc</code> ...	24
New <code>bwdistgeodesic</code> Function Computes Geodesic Distance Transform .....	25

New graydist Function Computes Gray-Weighted Distance Transform .....	26
New imapplymatrix Function Computes Linear Combination of Color Channels .....	27
Performance Improvements .....	28
hdrread Now Correct for Small Values .....	29
Change in Behavior for dicomwrite .....	30
Warning and Error ID Changes .....	31
Functions and Function Elements Being Removed .....	32

## **R2011a**

New bwconvhull Function Computes Convex Hull Image .....	38
New dicomwrite Option Writes Multiframe Imagery to Single File .....	39
nitfread Now Reads NITF Files with JPEG-Compressed Images .....	40
Reduced Memory Use for std2 .....	41
Reduced Memory Use for watershed .....	42
iccread and iccwrite Now Warn in Cases of Unrecognized PrimaryPlatform Signatures .....	43
Plot Selector Now Includes implay .....	44
Support for Code Generation from MATLAB .....	45
edge Function No Longer Smooths Image Twice .....	46
Functions and Function Elements Being Removed .....	47

## **R2010b**

New corner Function Detects Corners in Image .....	50
Efficient Display and Navigation of Very Large Images of Arbitrary Format in imtool .....	51
Now Possible to Control Padding Behavior when Using the blockproc Function .....	52
Writing to JPEG2000 File Format Supported by blockproc .....	53
Enhancements to the dicomread Function .....	54
The ImageMagnification Field of the nitfinfo Function Now Returns a Numeric Value .....	55

Performance Improvements .....	56
Functions and Function Elements Being Removed .....	57

## **R2010a**

New ImageAdapter Class Supports Custom File Formats for blockproc .....	60
The blockproc Function Now Supports Spatially Varying Operations .....	61
Plot Selector Now Generates Plots for imshow and imshow .....	62
makeform Now Supports White Point Adaptation .....	63
Intel Integrated Performance Primitives Library Support Extended to imdilate, imerode, and medfilt2 .....	64
imreconstruct Now Supports int64 and uint64 .....	65
Performance Improvements .....	66
Non-interactive Syntax of improfile Returns Different Output .....	67

## **R2009b**

New blockproc Function to Process Large Images .....	70
Intel Integrated Performance Primitives Library Upgraded and Support Extended to maci64 .....	72
Expanded hough Function Allows Specification of Arbitrary Theta Search Space .....	73
The imfilter Function Now Faster for uint16 and double Inputs .....	75
Improved Speed for Calculating N-D Euclidean Distance Transforms with the bwdist Function .....	76
Modified Behavior for the regionprops ConvexHull Property .....	77
Efficient Display and Navigation of Very Large NITF-File Images in imshow .....	78
Performance Improvements .....	79

## **R2009a**

Faster, Less Memory-Intensive Workflow for Labeling Regions and Measuring Their Properties .....	82
Multithreaded Implementation of imfilter Function .....	83
Efficient Display and Navigation of Very Large Images in imtool .....	84
New Dialog Box for Setting Toolbox Preferences .....	85
New imcolormaptool Function That Opens Choose Colormap Tool .....	86
End Point and Branch Point Detection Now Possible .....	87
nitfread Now Allows Image Subregion Selection .....	88
Support for Intel IPP on Mac .....	89
getColor, getLabelVisible, and setLabelVisible Methods Added to imdistline .....	90
Five Functions Moved to MATLAB .....	91
Fan-Beam Functions Updated .....	92

## **R2008b**

Performance Improvements .....	94
New cornermetric Function Detects Corners .....	95
Now Support Absolute Colorimetric Rendering Intent for GrayTRC and MatTRC .....	96
New createMask Method Creates Mask for Any ROI .....	97
Interactive Tools Refresh when Target Image Changes ...	98
The imscrollpanel 'PreserveView' Parameter Now Works for Images of All Sizes .....	99
Distance Tool and Cropping Tool Now Modes in imtool ...	100
In imtool Opening Adjust Contrast Tool No Longer Selects Window/Level Tool .....	101
immovie Command No Longer Shows Preview .....	102
Replace Calls to ipttable Function with MATLABuitable Function .....	103
imcontour Second Output Argument Changed .....	105
impixelinfo Tool Disappears when Image Changes .....	106
Some Code Moved into Different Directories .....	107
Functions and Demos Being Removed .....	108

## **R2008a**

Create High Dynamic Range (HDR) Images and Write Them to Files .....	110
Measure Properties of Regions in Grayscale Images .....	111
Display Very Large Images by Subsampling .....	112
Enhancements to ROI Tools .....	113
Enhancements to Color Functions .....	115
cp2tform Function Supports New Transformations .....	116
hough Function Uses Specified RhoResolution Values .....	117
Enhancements to Interactive Tools .....	118
New and Updated Demos .....	119
Enhancements to Other Functions .....	120

## **R2007b**

New Interactive Image Sequence and Video Viewer .....	122
Image Tool Includes Cropping, Enhanced Contrast Adjustment, and Saving of Modified Images .....	123
New Function for Converting Bayer Pattern Encoded Images to RGB .....	124
New Function for Creating a Multiresolution Gaussian Pyramid .....	125
Enhanced ROI Definition Behavior for imcrop, roifill, and roipoly .....	126
New Modular Interactive GUI-building Tools .....	127
New Programmable ROI Tools .....	128
Support for Reading NITF and HDR Images .....	129
Enhanced Performance .....	130
DICOM Dictionary Upgrade .....	131
Changes to Other Functions .....	132

## **R2007a**

Enhancements to imresize Function .....	134
applycform Supports Tetrahedral Interpolation .....	135
Control Point Selection Tool Enhancements .....	136
Enhancements to impoint, imline, and imrect Functions ..	137
Enhancements to montage Function .....	138

Changes to Other Functions .....	140
----------------------------------	-----

### **R2006b**

Enhancements to DICOM Capabilities .....	142
New Symmetric Option with graycomatrix Function .....	143
Enhancements to ICC Color Capabilities .....	144
Enhancements to the imdistline Function .....	145
setColor Method Accepts Predefined Color Strings .....	146

### **R2006a**

Enhanced ICC Profile Capabilities .....	148
New Pointer Management Functions .....	149
New Constraint Creation Function .....	150
Functions cp2tform, tforminv, imtransform .....	151
IPPL Not Used on 64-Bit Systems .....	152

### **R14SP3**

Support for Two New Medical Image File Formats .....	154
New Point, Rectangle, and Line Functions .....	155
Image Tool Enhancements and Improvements .....	156
New Utility Functions for Use with Profile-Based Color Space Conversion Functions .....	157
New Documentation on Processing Image Sequences .....	158
Control Point Selection Tool Now Works on Macintosh Systems .....	159
Obsolete and Deleted Functions .....	160
Image Tool is Not Compilable .....	161



---

Major Bug Fixes .....	164
New Directory Needed .....	169



# R2012b

---

Version: 8.1  
New Features: Yes  
Bug Fixes: Yes

## **Image gradient computation with `imgradient` and `imgradientxy` functions**

The toolbox includes two new functions for computing image gradients: `imgradient` and `imgradientxy`. `imgradient` computes the gradient magnitude and direction. `imgradientxy` computes the X and Y directional gradients

## **Histogram matching with `imhistmatch` function**

The new function `imhistmatch` adjusts the histogram of an image to match the N-bin histogram of a reference image.

## **Multilevel thresholding with `multithresh` and `imquantize` functions**

The new `imquantize` and `multithresh` functions enable multilevel grayscale thresholding and labeling. `imquantize` labels an image using a fixed range of grayscale levels. `multithresh` computes threshold levels using Otsu's method. The levels computed by `multithresh` can be used as input to `imquantize`.

## **3-D image registration with `imregister` function**

In addition to 2-D images, `imregister` can now align 3-D images using intensity-based registration of the images' voxel data.

## **Code generation for bwmorph and bwlookup with MATLAB Coder**

Standalone C code can be generated for bwmorph and bwlookup. The generated C code meets the strict memory and data type requirements of embedded target environments. To generate this code you need a MATLAB Coder license.



## **Added function bwlookup**

**Compatibility Considerations: Yes**

### **Compatibility Considerations**

Function applylut is not recommended; use bwlookup instead.

## **Writing private metadata when anonymizing DICOM files**

The `dicomanon` function now supports writing private metadata fields using the `'WritePrivate'` parameter

## Expanded color options with `imshowpair`

`imshowpair` now supports additional color options for displaying image differences and stereo imagery when using the new `'ColorChannels'` parameter. Using the `'red-cyan'` value with this parameter is particularly useful for viewing stereo anaglyphs.

## **Performance improvements**

The performance of the following functions has improved by taking advantage of hardware optimizations and multicore capabilities:

- `adapthisteq`
- `histeq`
- `imrotate`
- `intlut`

The `dicomanon`, `dicominfo`, `dicomlookup`, `dicomread`, `dicomwrite` functions have optimized implementations.

## **New Example**

Detect and measure circular objects in an image (`ipexcircles`).



# R2012a

---

Version: 8.0  
New Features: Yes  
Bug Fixes: Yes

## **Intensity-Based Image Registration**

The new `imregister` function lets you automatically align two images using intensity values, even when the images were created by two different devices (multimodal). With intensity-based registration, you do not need to specify control points.

You use the new `imregconfig` function to create the optimizer and the metric that `imregister` uses to specify the desired registration parameters.



## Two New Functions to Visually Compare Images

The toolbox includes two new functions for visually comparing images: `imshowpair` and `imfuse`.

`imshowpair` creates a composite of two images and displays them in a figure.

`imfuse` creates a composite of two images and returns a third image that is a numeric matrix containing a fused version of the original images.

## Circle Detection Using the Circular Hough Transform

The new `imfindcircles` function uses the Hough transform to find circular elements in grayscale, RGB, or binary images. To view the circles that have been detected, overlaid on the original image, use the `viscircles` function.

## **Performance Improvements**

The performance of the `imlincmb` function has improved by taking advantage of multicore capabilities.

## **New and Updated Demos**

The toolbox includes these new and updated demos.

- Registering Multimodal MRI Images (`ipexknee`)
- Finding the Rotation and Scale of a Distorted Image (`ipexrotate`)
- Measuring the Radius of a Roll of Tape (`ipexradius`) – Updated to use the new `imfindcircles` function

## **Data Type Change to Output Variable for `bwdist`**

### **Compatibility Considerations: Yes**

The `bwdist` function returns two output variables: the Euclidean distance transform and the closest-pixel map. The data type of the second output variable, the closest-pixel map, has changed.

### **Compatibility Considerations**

Before R2012a, the data type of the second output variable returned by the `bwdist` function was `single`. Now the data type of the second output variable returned by `bwdist` is dynamically chosen.

## **iradon Function Updated**

### **Compatibility Considerations: Yes**

A small imprecision in the way the dc level was computed in `iradon` has been corrected.

### **Compatibility Considerations**

Output variables `I` and `H` will be slightly different (about 1% to 2%) than in previous versions.

## Functions Being Removed

Compatibility Considerations: Yes

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
iptcheckinput	Still works but issues a warning	validateattributes	Replace all existing instances of iptcheckinput with validateattributes.
iptcheckstrs	Still works but issues a warning	validatestring	Replace all existing instances of iptcheckstrs with validatestring.
iptchecknargin	Still works but issues a warning	narginchk	Replace all existing instances of iptchecknargin with narginchk.





# R2011b

---

Version: 7.3  
New Features: Yes  
Bug Fixes: Yes

## **Parallel Block Processing Now Possible with blockproc**

If you have Parallel Computing Toolbox™, you can now use a new option in `blockproc` to improve performance of block processing tasks. Set the `'UseParallel'` argument to `true` to use this option.

## **New `bwdistgeodesic` Function Computes Geodesic Distance Transform**

Use `bwdistgeodesic` to compute the geodesic distance transform for a binary image.

## **New graydist Function Computes Gray-Weighted Distance Transform**

Use `graydist` to compute the gray-weighted distance transform of a grayscale image.

## **New `imapplymatrix` Function Computes Linear Combination of Color Channels**

The new `imapplymatrix` function applies a weighted sum to the color planes of an image for use in color space conversions.

## **Performance Improvements**

### **Faster Functions**

- `imhist` for large images
- `rgb2gray`

## **hdrread Now Correct for Small Values**

**Compatibility Considerations: Yes**

The `hdrread` function now returns correct small values. Specifically, the special case value of `(0,0,0,0)` now maps to `(0,0,0)`.

### **Compatibility Considerations**

Before R2011b, the value `(0,0,0,0)` mapped to `(0.5740E-41, 0.5740E-41, 0.5740E-41)`.

## **Change in Behavior for dicomwrite**

### **Compatibility Considerations: Yes**

If you use the `dicomwrite` function with the `'CreateMode'` option set to `'Create'` and pass in a data structure that contains the `'InversionTime'` tag, you will always receive `'InversionTime'` in your output.

### **Compatibility Considerations**

Before R2011b, if your input structure contained the `'InversionTime'` field, the DICOM file may or may not contain the `'InversionTime'` field. The inclusion of `'InversionTime'` depended on other parameters.



## **Warning and Error ID Changes**

### **Compatibility Considerations: Yes**

Many warning and error IDs have changed from their previous versions. These warnings or errors typically appear during a function call.

### **Compatibility Considerations**

If using warning or error IDs, you might need to change the strings you use. For example, if you turned off a warning for a certain ID, the warning might now appear under a different ID. If you use a `try/catch` statement in your code, replace the old identifier with the new identifier. There is no definitive list of the differences, or of the IDs that changed.

**Functions and Function Elements Being Removed**  
**Compatibility Considerations: Yes**

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
edge function — K-direction syntax	Errors	<code>BW = edge(..., direction)</code>	The syntax <code>BW = edge(...,K)</code> has been removed. Use the <code>BW = edge(...,direction)</code> syntax instead.
edge function — marr-hildreth syntax	Errors	<code>edge(I, 'marr-hildreth', ...)</code>	The syntax <code>edge(I, 'marr-hildreth', ...)</code> has been removed. Use the <code>edge(I, 'log', ...)</code> syntax instead.
imfeature	Errors	regionprops	imfeature has been removed. Use regionprops instead.
immovie — <code>immovie(D,size)</code>	Errors	<code>immovie(X,map)</code>	<code>immovie(D,size)</code> is an obsolete syntax and is no longer supported. Use <code>immovie(X,map)</code> instead.
imrotate function — [R,G,B] output syntax	Errors	<code>RGB2 = imrotate(RGB1)</code>	The syntax <code>[R,G,B] = imrotate(RGB)</code> has been removed. Use the <code>RGB2 = imrotate(RGB1)</code> syntax instead.
imrotate — no output argument syntax	Starting in R2011b, no output argument syntax returns result in ans.	Call <code>imshow</code> to display the output of <code>imrotate</code>	Replacement is to call <code>imshow</code> to display the output of <code>imrotate</code> , like this: <code>B = imrotate(A,30); imshow(B)</code>

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
imshow function — imshow(..., DISPLAY_OPTION)	Errors	<pre>imshow(..., 'InitialMagnification', 100)</pre> <p>or</p> <pre>imshow(..., 'InitialMagnification', 'fit')</pre>	<ul style="list-style-type: none"> <li>The syntax <code>imshow(...,'truesize')</code> has been removed. Use the <code>imshow(...,'InitialMagnification',100)</code> syntax instead.</li> <li>The syntax <code>imshow(...,'notruesize')</code> has been removed. Use the <code>imshow(...,'InitialMagnification','fit')</code> syntax instead.</li> </ul>
imshow function — imshow(x,y,...) syntax	Errors	<pre>imshow(...,'XData', x,'YData',y)</pre>	The syntax <code>imshow(x,y,...)</code> has been removed. Use the <code>imshow(...,'XData',x,'YData',y)</code> syntax instead.
imshow function — imshow(I,N) syntax	Errors	Not applicable	The syntax <code>imshow(I,N)</code> has been removed. Your grayscale image will be displayed using 256 shades of gray.
imview	Errors	imtool	imview has been removed. Use imtool instead.
iptsetpref function — 'ImshowTruesize' preference	Errors	'ImshowInitialMagnification'  preference	Replace all existing instances of 'ImshowTruesize' with 'ImshowInitialMagnification'.

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
iptsetpref function — 'ImviewInitialMagnification' preference	Errors	'ImtoolInitialMagnification' preference	Replace all existing instances of 'ImviewInitialMagnification' with 'ImtoolInitialMagnification'.
Any calls to iptsetpref from within a startup.m file	It will do nothing.	Any calls to iptsetpref in any session are persistent.	You can safely remove calls to iptsetpref from startup.m files.
isbw	Errors	Not applicable	isbw has been removed. No replacement.
isgray	Errors	Not applicable	isgray has been removed. No replacement.
isind	Errors	Not applicable	isind has been removed. No replacement.
isrgb	Errors	Not applicable	isrgb has been removed. No replacement.
medfilt2 function — medfilt2(A,[M N],[Mb Nb],...) syntax	Errors	Not applicable	No replacement.
montage — montage(D,[M N P]) syntax	Errors	Not applicable	Syntax has been removed. No replacement.

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
uintlut	Errors	intlut	uintlut has been removed. Use intlut instead.
wiener2 function — wiener2(I,[m n],[mblock nblock],...) syntax	Errors	wiener2(I,[m n]) or wiener2(I,[m n],noise)	<p>The syntax wiener2(I,[m n],[mblock nblock]) has been removed. Use the wiener2(I,[m n]) syntax instead.</p> <p>The syntax wiener2(I,[m n],[mblock nblock],noise) has been removed. Use the wiener2(I,[m n],noise) syntax instead.</p>

# R2011a

---

Version: 7.2  
New Features: Yes  
Bug Fixes: Yes

## **New `bwconvhull` Function Computes Convex Hull Image**

Use `bwconvhull` to compute the convex hull image from a binary image.



## **New dicomwrite Option Writes Multiframe Imagery to Single File**

Set the new 'MultiframeSingleFile' option of the dicomwrite function to true to write multiframe imagery to one file, regardless of how many frames the input image contains.

## **nitfread Now Reads NITF Files with JPEG-Compressed Images**

If you have NITF files containing JPEG-compressed images, you can now read them using `nitfread`.

## Reduced Memory Use for std2

### Compatibility Considerations: Yes

The `std2` function has improved performance for large 1-, 8-, and 16-bit integers.

### Compatibility Considerations

Compared to releases before R2011a, the `std2` function now returns slightly different results for some images. To receive the same results as previously, use this code:

```
% For input image im
if ~isa(im,'double')
    im = double(im);
end
std_old = std(im(:));
```

## **Reduced Memory Use for watershed**

### **Compatibility Considerations: Yes**

The watershed function is now more memory efficient than it was in releases before R2011a.

### **Compatibility Considerations**

The watershed regions in the label matrix returned by watershed have different indices than they did before R2011a.

Also, the label matrix returned by watershed was class `double` in previous releases, and is now an unsigned integer class.

If you want to return a label matrix of class `double`, as you did before, use the `double` function to convert it:

```
L = watershed(A);  
L = double(L);
```

## **iccread and iccwrite Now Warn in Cases of Unrecognized PrimaryPlatform Signatures**

If `iccread` or `iccwrite` encounter an unrecognized PrimaryPlatform signature in the profile header, they will warn. In releases before R2011a, these functions would error instead of warn in cases with unrecognized PrimaryPlatform signatures.

## **Plot Selector Now Includes `implay`**

The `implay` function has been added to the list of functions available in the Plot Selector. You can now display data in `implay` directly from the Plot Selector workspace tool. For details about the Plot Selector, see [Enhanced Plot Selector Simplifies Data Display](#).

## Support for Code Generation from MATLAB

You can now generate standalone C code for two Image Processing Toolbox™ functions: `label2rgb` and `fspecial`. The generated C code meets the strict memory and data type requirements of embedded target environments. To generate this code you need a MATLAB Coder license. See the Code Generation for Image Processing Toolbox Functions chapter in the User's Guide for details, including limitations.

## **edge Function No Longer Smooths Image Twice**

**Compatibility Considerations: Yes**

In previous releases, the implementation of the Canny filter, called with the `canny` method of the `edge` function, smoothed the image twice while constructing the gradient image. The function smoothed the image once using a Gaussian filter and then used a first derivative of a Gaussian filter to extract a smoothed version of the image gradient. Smoothing an image and then differentiating it is the same as convolving the image with a derivative of the smoothing kernel, so this implementation had the effect of smoothing the image twice. In addition, the original implementation of the Canny filter included an extra morphological thinning step that is not in the published algorithm.

### **Compatibility Considerations**

The `edge` function no longer smooths an image twice. If you are setting the value of `sigma` and want similar results to the previous implementation, increase `sigma` by a factor of  $\sqrt{2}$ .

To achieve the same results produced by the previous implementation, use this syntax:

```
BW = edge(I, 'canny_old', ...)
```



## Functions and Function Elements Being Removed

Compatibility Considerations: Yes

Function or Function Element	What Happens When You Use This Function or Element?	Use This Instead	Compatibility Considerations
ipttable	Errors	uitable	Replace all existing instances of ipttable with uitable.



# R2010b

---

Version: 7.1  
New Features: Yes  
Bug Fixes: Yes

## **New corner Function Detects Corners in Image**

### **Compatibility Considerations: Yes**

The new `corner` function detects corners in a grayscale or binary image. Corners are a feature you can use to find the correspondence between images.

### **Compatibility Considerations**

In R2008b and later releases, you could find corners by computing a `cornermetric` matrix with the `cornermetric` function and then finding peak values. Now, you can simplify your workflow by using the `corner` function.

## **Efficient Display and Navigation of Very Large Images of Arbitrary Format in imtool**

The `rsetwrite` function allows you to create a multiresolution image pyramid (R-Set) from a large image file. In previous releases, the `rsetwrite` function accepted TIFF or NITF image files. Now, in addition to accepting these image files, `rsetwrite` accepts `ImageAdapter` objects. `ImageAdapter` objects allow you to work with images of arbitrary file format. See [Working with Data in Unsupported Formats](#) for guidelines on how to create an `ImageAdapter` object.

## **Now Possible to Control Padding Behavior when Using the `blockproc` Function**

With the new 'PadMethod' option, you can now control padding behavior when using the `blockproc` function. In previous releases, the `blockproc` function only supported zero padding along the boundary of the image. Now, the function supports padding the image with a scalar pad value, repeated border elements of A, or the mirror reflection of A.

## **Writing to JPEG2000 File Format Supported by blockproc**

The `blockproc` function now provides more flexibility in format choices. The function supports writing to JPEG2000 file formats with `*.jp2`, `*j2c`, or `*.j2k` extensions.

## **Enhancements to the dicomread Function**

The `dicomread` function has been enhanced in two ways: the function now reads multiframe imagery faster, and it reads files containing JPEG-2000 encoded imagery.



## **The ImageMagnification Field of the nitfinfo Function Now Returns a Numeric Value**

The ImageMagnification field of the nitfinfo function has been updated. Previously, if you used the function to return a structure with file-level metadata, the ImageMagnification field of the structure contained an incorrect value. (The incorrect value was either an empty image magnification value or the text value for the field.) Now, the ImageMagnification field returns the value for the image magnification.

## **Performance Improvements**

### **Faster Functions**

- `blockproc`
- `imresize`
- `iradon`

## Functions and Function Elements Being Removed

Compatibility Considerations: Yes

Function or Function Element Name	What Happens When You Use This Function or Element?	Use This Instead	Compatibility Considerations
blkproc	Still runs	blockproc	None



# R2010a

---

Version: 7.0  
New Features: Yes  
Bug Fixes: Yes

## **New ImageAdapter Class Supports Custom File Formats for blockproc**

The `blockproc` function, introduced in R2009b, supported file-based block processing for arbitrarily large images. In R2009b, you could use `blockproc` to read or write TIFF images or to read JPEG2000 images. Now, with the addition of the new `ImageAdapter` class, you can design your own class to use `blockproc` with images of arbitrary file format.

## **The blockproc Function Now Supports Spatially Varying Operations**

Additional fields have been added to the blockproc “block struct” that contain spatial information. These new fields facilitate operations that depend on location.

## **Plot Selector Now Generates Plots for `imshow` and `imshow`**

The Plot Selector workspace tool creates graphs of workspace variables. The `imshow` and `imshow` functions have been added to the list of possible plotting functions available in the Plot Selector. For more information about the Plot Selector, see [Enhanced Plot Selector Simplifies Data Display](#).



## makecform Now Supports White Point Adaptation

`makecform` uses the white point specified by the International Color Consortium (ICC) as the default for the `srgb2lab` and `lab2srgb` transform types. You can now adapt to a white point other than `whitepoint('ICC')`, the default value, by using a new syntax to specify the adapted white point:

```
C = makecform(type, 'AdaptedWhitePoint', WP)
```

You can also create a linear chromatic-adaptation transform:

```
C = makecform('adapt', 'WhiteStart', WPS, 'WhiteEnd', WPE, ...  
             'AdaptModel', modelName)
```

This transform allows you to adapt *XYZ* color values from one white point to another.

## **Intel Integrated Performance Primitives Library Support Extended to `imdilate`, `imerode`, and `medfilt2`**

The functions `imdilate` and `imerode` are now hardware optimized for `ones(3)` neighborhoods for `single`, `uint8`, and `uint16` input images.

The `medfilt2` function is now hardware optimized for integer data types (`uint8`, `uint16`, and `int16`) and the `single` data type with kernel size `3 x 3`.

## **imreconstruct Now Supports int64 and uint64**

The `imreconstruct` function now supports data types `int64` and `uint64`.

## Performance Improvements

### Faster Functions

- edge
- imdilate
- imerode
- imfilter
- imresize
- iradon
- medfilt2

### Multithreaded Functions

- bwmorph
- edge
- imabsdiff
- imadd
- imclose
- imdivide
- immultiply
- imopen
- iradon
- medfilt2

## Non-interactive Syntax of improfile Returns Different Output

### Compatibility Considerations: Yes

One of the non-interactive syntaxes of `improfile` now returns different output. The output for the syntax

```
C = improfile(I,xi,yi,N)
```

has changed. In the syntax above, `N` specifies the number of points for which to compute intensity values and `xi` and `yi` specify the spatial coordinates of the endpoints of the line segments.

For a given line defined by `xi` and `yi`, `improfile` now returns a profile sampled at both endpoints and all sampling points in between at roughly unit interval spacing. If the distance between `xi` and `yi` is `N` pixels, the profile is evaluated at `N+1` points.

### Compatibility Considerations

In previous releases, if you supplied the `xi` and `yi` end points as (1,1) and (10,1), the profile would be evaluated at nine points, the nine unit-length intervals between 1 and 10 in the continuous  $x$ - $y$  plane. These nine points would be the two end points, plus seven points in between.



# R2009b

---

Version: 6.4  
New Features: Yes  
Bug Fixes: Yes

## **New blockproc Function to Process Large Images**

### **Compatibility Considerations: Yes**

The new `blockproc` function supports file-based block processing for arbitrarily large TIFF images. The new function supports in-memory operations as well as file-to-file processing of images which are too large to load completely into memory.

### **Compatibility Considerations**

In previous releases, you could use the `blkproc` function for in-memory block-processing of images. The `blkproc` function will be removed in a future release. Replace all instances of `blkproc` with `blockproc`.

When updating your code from `blkproc` to `blockproc`, it is important to note that the user-defined function, `fun`, has a new signature. It now takes a structure, the "block struct," as input instead of simply a matrix of image data.

The examples below demonstrate how to update your code from `blkproc` to `blockproc`.

#### **Example One: DCT2**

```
% BLKPROC code
I = imread('cameraman.tif');
fun = @dct2;
J = blkproc(I,[8 8],fun);

% BLOCKPROC equivalent (using an anonymous function)
fun = @(block_struct) dct2(block_struct.data);
J = blockproc(I,[8 8],fun);
```

#### **Example Two: Filtering**

```
% BLKPROC code
I = imread('concordorthophoto.png');
h = fspecial('gaussian',[11 11],2.5);
fun = @(x) imfilter(x,h,'conv','same');
J = blkproc(I,[500 500],[5 5],fun);
```



```
% BLOCKPROC equivalent (using an anonymous function)
fun = @(block_struct) imfilter(block_struct.data,h,'conv','same');
J = blockproc(I,[500 500],fun,'BorderSize',[5 5]);
```

## **Intel Integrated Performance Primitives Library Upgraded and Support Extended to maci64**

The Intel® Integrated Performance Primitives (Intel IPP) Library has been upgraded from Version 5.3.1 to Version 6.0 Update 1. Intel IPP Library support has been extended to 64-bit Intel-based Mac computers.

## Expanded hough Function Allows Specification of Arbitrary Theta Search Space

**Compatibility Considerations: Yes**

The hough function now yields faster results for narrower *theta* ranges due to the addition of a parameter/value pair for specifying *theta* values.

### Compatibility Considerations

In previous releases, the 'ThetaResolution' parameter controlled the *theta* values for the hough function. Now 'ThetaResolution' is being replaced by the new 'Theta' parameter.

### Function Elements Being Removed

Function and Syntax	What Happens When You Use the Function or Element?	Use This Instead	Compatibility Considerations
hough(BW, 'ThetaResolution', val)	Still runs	hough(BW, 'Theta', -90:val:(90-val))	Input parameter no longer recommended. Use new 'Theta' parameter.

Note that with the introduction of the 'Theta' parameter, not all abbreviated forms of 'ThetaResolution' will still work. In previous releases, if you entered the following syntax:

```
hough(BW, 'T', val)
```

'T' stood for 'ThetaResolution'. Now if you enter this same syntax, 'T' stands for the new 'Theta' parameter.

If you have old code that uses the 'ThetaResolution' parameter, please see the definition below:

Parameter	Description
'ThetaResolution'	Real scalar value between 0 and 90, exclusive, that specifies the spacing (in degrees) of the Hough transform bins along the <i>theta</i> axis. Default: 1.

For 'ThetaResolution',  $n_{\theta} = 2 * \text{ceil}(90 / \text{ThetaResolution})$ . *theta* angle values are in the range  $[-90, 90)$  degrees. If  $90 / \text{ThetaResolution}$  is not an integer, the actual angle spacing is  $90 / \text{ceil}(90 / \text{ThetaResolution})$ .

## The `imfilter` Function Now Faster for `uint16` and `double` Inputs

**Compatibility Considerations: Yes**

The `imfilter` function now runs faster with `uint16` and `double` inputs than in previous releases. This performance enhancement is due to the use of the Intel IPP Library with inputs of these types.

### Compatibility Considerations

Using the Intel IPP library for `uint16` images, poses no compatibility issues. The same is not true, however, for the `double` data type.

If an input image contains NaN values and a filtering kernel contains zero values, the `imfilter` function now gives different results when the Intel IPP library is enabled versus when it is disabled. If you want to preserve the behavior of previous releases, use `iptsetpref('UseIPPL', false)` to disable the Intel IPP library.

## **Improved Speed for Calculating N-D Euclidean Distance Transforms with the `bwdist` Function**

**Compatibility Considerations:** Yes

A new algorithm improves the speed and reduces the memory footprint for the `bwdist` function.

### **Compatibility Considerations**

In previous releases, the `bwdist` function used different algorithms for computing the Euclidean distance transform and the associated label matrix. If you need the same results produced by the previous implementation, use the function `bwdist_old`.

## **Modified Behavior for the regionprops ConvexHull Property**

### **Compatibility Considerations: Yes**

The 'ConvexHull' property of `regionprops` depends on the MATLAB® `convhull` function. Due to changes in `convhull`, the results returned by 'ConvexHull' will now be slightly different than in previous releases.

### **Compatibility Considerations**

The order of the vertices returned by the 'ConvexHull' property of `regionprops` may differ from that returned in releases before R2009b. Also, the returned hull may contain additional collinear points that were omitted in previous releases.

## **Efficient Display and Navigation of Very Large NITF-File Images in imtool**

The `rsetwrite` function allows you to create multi-resolution image pyramids (R-Sets) that you can open in `imtool`. In previous releases, `rsetwrite` worked only with TIFF files. Now it accepts NITF files, as well, as long as they are Version 2.0 or greater, contain an uncompressed image, have integer data (no floating point data), and have three or fewer image bands. Finally, if a NITF file has more than one band of data, the data must be unsigned.



## Performance Improvements

The performance of several existing toolbox functions has been improved in this release. In some cases, other toolbox functions call these functions and therefore will also benefit from these speed improvements.

### Faster Functions

- `bwdist`
- `imcomplement`
- `imdilate`
- `imerode`
- `imfilter`
- `improfile`
- `imrotate`

### Multithreaded Functions

- `applylut`
- `bwpack`
- `bwunpack`
- `imdilate`
- `imerode`
- `imreconstruct`



# R2009a

---

Version: 6.3  
New Features: Yes  
Bug Fixes: Yes

## **Faster, Less Memory-Intensive Workflow for Labeling Regions and Measuring Their Properties**

The `bwconncomp` function computes connected components for binary images. It uses significantly less memory and is sometimes faster than `bwlabel` and `bwlabeln`.

To extract features from a binary image using `regionprops` with default connectivity, just pass `BW` directly into `regionprops` (i.e., `regionprops(BW)`). To compute a label matrix having more memory-efficient data type (e.g., `uint8` versus `double`), use the `labelmatrix` function on the output of `bwconncomp`.

## **Multithreaded Implementation of imfilter Function**

The `imfilter` function is now multithreaded.

## **Efficient Display and Navigation of Very Large Images in `imtool`**

The new `rsetwrite` function allows you to create a multi-resolution image pyramid (R-Set) from a large TIFF image file. In previous releases, large images would not open in `imtool`, or they did open, but navigation was slow. You can now open your R-Set with `imtool` and explore it as you would a standard image.

## **New Dialog Box for Setting Toolbox Preferences**

### **Compatibility Considerations: Yes**

A new preferences dialog box allows customization of Image Processing Toolbox preferences. You can access the dialog box via the **File** menu in the MATLAB desktop, the **File** menu in the Image Tool (`imtool`), or directly from the command line by typing `iptprefs`.

A new preference has been added that allows you to specify whether the Overview tool opens automatically when you launch the Image Tool.

### **Compatibility Considerations**

In previous releases, the Overview tool opened automatically with `imtool`. The new default behavior is for the Overview tool to no longer open automatically. If you would like to revert to the previous behavior you can set this preference via the Image Processing Preferences dialog box (`iptprefs`).

In previous releases, if you changed the preferences with the `iptsetpref` command, these changes would revert to the default setting when you finished a MATLAB session. Now, if you change preferences, these changes will remain intact from one MATLAB session to the next.

## **New imcolormaptool Function That Opens Choose Colormap Tool**

The new function `imcolormaptool` opens the Choose Colormap tool. The Choose Colormap tool allows you to interactively change the colormap of a displayed image. You can also access the tool from the **Tools** menu of the Image Tool, as in previous releases.



## **End Point and Branch Point Detection Now Possible**

`bwmorph` now detects end points and branch points in binary images.

## **nitfread Now Allows Image Subregion Selection**

### **Compatibility Considerations: Yes**

The `nitfread` function now includes a `PixelRegion` parameter that returns a subimage as specified by row and column vectors.

### **Compatibility Considerations**

From R2007a to R2008b, the `nitfread` function returned `uint8` data for images with 1-bit data. Now `nitfread` returns `logical` data for images with 1-bit data. If you want this function to behave as it did in the past, enter the following:

```
imdata = uint8(nitfread(filename));
```

## **Support for Intel IPP on Mac**

In previous releases, the Image Processing Toolbox leveraged the Intel Integrated Performance Primitives (Intel IPP) Library on 32- and 64-bit Linux® and Windows® platforms. Now Intel IPP-use has been extended to the Mac.

## **getColor, getLabelVisible, and setLabelVisible Methods Added to imdistline**

`imdistline` now includes a `getColor` method that returns the color used to draw a specific ROI object. Also, the new `getLabelVisible` and `setLabelVisible` methods make it possible to control the visibility of the Distance tool text label.

## Five Functions Moved to MATLAB

### Compatibility Considerations: Yes

The following five functions moved from the Image Processing Toolbox to MATLAB: `cmpermute`, `cmunique`, `dither`, `imapprox`, and `rgb2ind`. The behavior of some of the functions has changed slightly, as described in the compatibility considerations listed below.

### Compatibility Considerations

- Functions `dither` and `imapprox`, when called without output arguments, no longer display their output as an image via a call to `imshow`. Now, if you want to display the resulting image, assign the output to one or more variables and call `imshow`. For example, try the following:

```
[Y,newmap] = imapprox(X,map,n)
imshow(Y,newmap)
```

- Function `rgb2ind` errors when called with the syntax `rgb2ind( RGB )`. You must specify the number of colors, tolerance, or colormap. For example, you can use the syntax `rgb2ind( RGB, 128 )`, where 128 represents the number of colors.
- Function `imapprox` errors when called with the syntax `imapprox( x, map )`. As with `rgb2ind`, you must specify additional parameters.

## **Fan-Beam Functions Updated**

**Compatibility Considerations: Yes**

### **Compatibility Considerations**

Due to a bug fix, the fan-beam functions (`fanbeam`, `ifanbeam`, `fan2para`, `para2fan`) now return different answers than in previous releases.

# R2008b

---

Version: 6.2  
New Features: Yes  
Bug Fixes: Yes

## Performance Improvements

The performance of several existing toolbox functions has been improved in this release, including:

- Binary erosion and dilation (`imdilate`, `imerode`, `bwhitmiss`, and `rangefilt`)
- `graycomatrix`
- Image arithmetic and filtering now leverage the IPP Library on 32- and 64-bit Windows and Linux platforms.



## **New cornermetric Function Detects Corners**

New cornermetric function detects corners.

## **Now Support Absolute Colorimetric Rendering Intent for GrayTRC and MatTRC**

New additions to the `makecform` syntax include rendering intents for the Matrix/Tone Reproduction Curve (MatTRC) model and the single-channel Tone Reproduction Curve (GrayTRC) model.

## **New createMask Method Creates Mask for Any ROI**

Use the new `createMask` method of the `imroi` base class to return a mask, or binary image, that is the same size as the input image with 1s inside the ROI object and 0s outside. The new method is available in the following classes: `impoint`, `imline`, `imrect`, `imellipse`, `impoly`, and `imfreehand`.

## **Interactive Tools Refresh when Target Image Changes**

The following modular interactive tools now update automatically if you modify the target image: Adjust Contrast, Pixel Region, Pixel Information, Overview, Display Range, and Image Information.

## **The imscrollpanel 'PreserveView' Parameter Now Works for Images of All Sizes**

**Compatibility Considerations: Yes**

The `replaceImage` function in the `imscrollpanel` API has been modified. You can now use the `'PreserveView'` parameter even in cases where your replacement image is not the same size as your original image. The new image will appear with the center of view in the same relative position as in the original image.

### **Compatibility Considerations**

In previous releases, the default for different size images was for the new image to appear centered and at 100% magnification.

## **Distance Tool and Cropping Tool Now Modes in imtool**

### **Compatibility Considerations: Yes**

The Distance tool and Cropping tool have been modified. Now to use the Distance tool, you click one end of the distance to be measured, drag, and release to complete the measurement. With the new version of the Cropping tool, you may click and drag to define the cropping region as many times as you want. If you define one region and then decide to crop a different region instead, simply click and drag the mouse again to define the new region.

### **Compatibility Considerations**

In previous releases, the Distance tool appeared as a horizontal bar of set length. You could drag the ends of the tool to change size and orientation.

In previous releases, if you defined one cropping region, you could move this box or change the size, but you couldn't start with a new box unless you canceled the tool, clicked on the "Crop Image" toolbar button, and then defined the new region.

## **In imtool Opening Adjust Contrast Tool No Longer Selects Window/Level Tool**

**Compatibility Considerations: Yes**

When you open the Adjust Contrast tool, the Window/Level tool is no longer turned on automatically. To operate this feature, simply select the Window/Level tool icon from the Image Tool toolbar. (To identify the icon, note that if you move the cursor over the Window/Level tool icon, the words “Adjust contrast/brightness via mouse motion” appear.) Or, you can select “Tools” from the Image Tool menu and then click on “Window/Level.”

### **Compatibility Considerations**

In previous releases, when you opened the Adjust Contrast tool, the Window/Level tool automatically turned on at the same time. Note that the Window/Level tool still turns on when you call `imcontrast` from the command line.

## **immovie Command No Longer Shows Preview**

### **Compatibility Considerations: Yes**

`immovie` no longer opens a figure window to display the movie as it is being created. You can display and explore the output of `immovie` using `implay`.

### **Compatibility Considerations**

If you want to use `movie` to visualize the output but don't know how to set up the figure appropriately, call `imshow` on one of the movie frames first before calling `movie`.



## Replace Calls to ipttable Function with MATLAB uitable Function

### Compatibility Considerations: Yes

The ipttable function is being deprecated and will be removed in a future release.

### Compatibility Considerations

If you used the ipttable function to display tabular data, you should replace use of ipttable with the MATLAB function uitable.

If you used the cell array syntax of ipttable, make the following changes to your code to achieve a similar effect using uitable. For more information about using uitable, see the uitable function reference page.

R2008a Code	R2008b Code
<pre>table = ipttable(parent,cell_array_data);</pre>	<pre>table = uitable(parent,'Data',cell_array_data);</pre>

If you used the struct syntax of ipttable, make the following changes to your code to achieve a similar effect with uitable.

R2008a Code	R2008b Code
<pre>table = ipttable(parent,struct_data);</pre>	<pre>field_names = fieldnames(struct_data); values = struct2cell(struct_data); for idx = 1:numel(values)     val = values{idx};     if ~ischar(val)    size(val,1) &gt; 1         values{idx} = evalc('disp(values{idx})');     end end table = uitable(parent,'Data',     [field_names values]);</pre>

Code written in previous releases that depends on `iptables` will begin to warn and eventually error in later releases.

## **imcontour Second Output Argument Changed**

### **Compatibility Considerations: Yes**

The second output argument of `imcontour` is now a handle to an `hggroup` object instead of an array of handles to patch objects.

### **Compatibility Considerations**

If you need to access handles of individual patch objects, use the following code to work around the change.

```
[c, handleToHGGroup] = imcontour(..);  
arrayOfHandlesToPatchObjects = get(handleToHGGroup, 'Child');
```

## **impixelinfo Tool Disappears when Image Changes**

**Compatibility Considerations: Yes**

If you use `imshow` to display an image, open the `impixelinfo` tool, and use `imshow` to open a new image, the `impixelinfo` tool will disappear along with the first image.

### **Compatibility Considerations**

In previous versions, if you entered the following code:

```
imshow pout.tif
impixelinfo
imshow peppers.png
```

the `impixelinfo` tool would update to reflect changes to the image. Now you must call the `impixelinfo` tool again after opening the second image.

## **Some Code Moved into Different Directories**

- Colorspace functionality moved into the new `toolbox/images/colorspaces` directory.
- Medical file formats moved into the `toolbox/images/iptformats` directory with other file formats, and the `toolbox/images/medformats` directory was removed.

## Functions and Demos Being Removed

Compatibility Considerations: Yes

Function or Demo Name	What Happens When You Use Function or Demo?	Use This Instead	Compatibility Considerations
pixval	Errors	Use <code>impixelinfo</code> for pixel reporting and <code>imdistline</code> for measuring distance.	Replace all existing instances of <code>pixval</code> with <code>impixelinfo</code> or <code>imdistline</code> .
dctdemo	Errors	NA	None
edgedemo	Errors	NA	None
firdemo	Errors	NA	None
landsatdemo	Errors	NA	None
nrfiltdemo	Errors	NA	None
qtdemo	Errors	NA	None
roidemo	Errors	NA	None

# R2008a

---

Version: 6.1  
New Features: Yes  
Bug Fixes: Yes

## **Create High Dynamic Range (HDR) Images and Write Them to Files**

Create a high dynamic range image from a group of low dynamic range images using the new `makehdr` function. The low dynamic range images must be spatially registered. You can write the HDR image to a file using the `hdrwrite` function. These functions complement the `hdrread` and `tonemap` functions introduced in R2007b.



## **Measure Properties of Regions in Grayscale Images**

The `regionprops` function now accepts grayscale images as an input parameter, returning measurements based on the values of pixels in specified regions. Using `regionprops`, you can obtain measurements of regions in the image such as the maximum, minimum, and mean intensities in the region, and the weighted centroid.

## **Display Very Large Images by Subsampling**

You can now display very large images from TIFF files by using the `imshow` function's new 'Reduce' parameter. When you specify this parameter, `imshow` displays a subsampled version of the image. `imshow` determines the subsampling factor by considering the size of the image and the reduction required to fit the image on your screen. The 'Reduce' parameter makes it possible to view very large images in their entirety that could not previously be displayed. Note, however, that the image subsampling that is performed reduces that amount of image data displayed.

## Enhancements to ROI Tools

### Compatibility Considerations: Yes

The toolbox includes several functions that enable the definition of regions of interest of various shapes: `impoint`, `imline`, `impoly`, `imrect`, and `imfreehand`. These ROI tools have several enhancements:

### ROI Tools Reimplemented as MATLAB Classes

The ROI tools have been reimplemented as MATLAB classes. This change does not affect how the ROI tools function; they function identically to their previous implementation. The documentation uses the MATLAB functional syntax descriptions rather than the dot notation. That is, the documentation shows how to call the class methods specifying a handle to the object as the first argument, `method(h, ...)`. Note, however, that you can still use the dot notation when calling the methods, `obj.method(...)`. In addition, the `iptgetapi` function now returns an object of the new class which means that code similar to the following will continue to work:

```
api = iptgetapi(h)
api.method()
```

### Compatibility Considerations

The class of the data returned by the ROI tools is now a handle to an ROI class, such as `imline` or `impoly`. In addition, several undocumented methods supported by the ROI tools have been removed: `getContextMenu`, `setContextMenu`, `getDrawAPI`, `addCallback`, and `removeCallback`.

### ROI Tools Support New wait and resume Methods

The ROI tools now support `wait` and `resume` methods so that they can be used in scripts. By using the `wait` method, you can enable users of your script to make the initial placement of the ROI, adjust the ROI and accept it, and then use the position in the script. For example, using the `wait` method with an ROI tool, you could write a script that creates a mask.

The `resume` method is a programmatic way to return control to the command line. When called after `wait`, `resume` causes `wait` to return the accepted position of the ROI.

### **Interactively Add New Vertices to ROI Polygons**

You can now add vertices interactively to polygonal ROIs that you define using the `impoly` function. To create the new vertex, position the pointer over an edge of the polygon and press the A key. The pointer changes shape. Click the mouse to add a new vertex. The `roifill` and `roipoly` functions, which use `impoly` to implement ROIs, also support this new capability.

## Enhancements to Color Functions

The following color functions have been enhanced.

### **makecform Supports Converting Between sRGB and CMYK**

The `makecform` function now supports two new color space conversion types for converting between sRGB and CMYK: `'srgb2cmyk'` and `'cmyk2srgb'`.

### **iccwrite Creates Smaller ICC Profiles**

The `iccwrite` function now uses certain optimizations to reduce the size of the International Color Consortium (ICC) color profiles that it creates. `iccwrite` uses aliasing to avoid writing tag data multiple times when it is included in more than one profile table.

## **cp2tform Function Supports New Transformations**

**Compatibility Considerations: Yes**

The cp2tform function supports two new transformation types: 'similarity' and 'nonreflective similarity'.

### **Compatibility Considerations**

The 'linear conformal' transformation type supported by the cp2tform function has been renamed to 'nonreflective similarity'.

## **hough Function Uses Specified RhoResolution Values**

**Compatibility Considerations: Yes**

The hough function now uses the value you specify for the 'RhoResolution' parameter. In previous releases, the function did not use the value specified.

### **Compatibility Considerations**

The Hough matrix, H, and the Rho outputs returned by the hough function have different results than those obtained from the same function in previous releases.

## Enhancements to Interactive Tools

The following modular interactive tools have been enhanced.

- Adjust Contrast tool (`imcontrast`) — The **Adjust Data** button in the Adjust Contrast tool is disabled until you make a change to image contrast.
- Pixel Region tool (`impixelregion`) — To improve the visibility of the image pixels being examined, the Pixel Region tool stops including grid lines in the display at low magnifications.



## New and Updated Demos

The toolbox includes the following new and changed demos.

- *Batch Processing Image Files in Parallel* is an existing demo that has been updated, and simplified, through use of the `parfor` function.
- *Detecting Cars in a Video of Traffic* is a new demo that shows how to use the toolbox to visualize and analyze videos or image sequences.
- *Measuring Regions in Grayscale Images* is a new demo that shows how to use the `regionprops` function with grayscale images.

## Enhancements to Other Functions

This release includes changes to the following functions.

<b>Function</b>	<b>Description of Enhancement</b>
<code>imageinfo</code>	Accepts files of several additional file formats as an input argument, including NITF, Interfile, and Analyze file formats.
<code>imshow</code>	Supports a new colormap parameter for specifying a colormap for grayscale images.
<code>imtool</code>	Supports a new colormap parameter for specifying a colormap for grayscale images.
<code>truecolor</code>	Preserves the border preference setting of the figure when adjusting the image display size.

# R2007b

---

Version: 6.0  
New Features: Yes  
Bug Fixes: Yes


## **New Interactive Image Sequence and Video Viewer**

The toolbox now supports a new interactive image sequence viewer, called the Movie Player (`implay`). Using the Movie Player you can:

- Play a MATLAB movie, AVI file, or multidimensional array.
- Step through a movie or sequence of images, frame-by-frame, or jump to the beginning or end of the sequence.
- Examine a frame using the Pixel Region tool or export the frame to the Image Tool.

## **Image Tool Includes Cropping, Enhanced Contrast Adjustment, and Saving of Modified Images**

The Image Tool (imtool) supports several enhancements:

- You can now modify the image data after performing a contrast adjustment operation. Previously, contrast adjustment only affected the display of the image, not the actual image data. To modify image data, click **Adjust Data** in the Adjust Contrast tool.
- You can now interactively crop an image displayed in the Image Tool using the Crop Image button  in the toolbar (or select **Crop Image** from the Tools menu).
- You can now save the image displayed in the image tool in any of several common image file formats. Select **Save As** from the Image Tool File menu.

## **New Function for Converting Bayer Pattern Encoded Images to RGB**

The toolbox now supports a function, `demosaic`, that can convert a Bayer pattern encoded image into an RGB image.

A Bayer filter mosaic, or color filter array, refers to the arrangement of color filters that let each sensor in a single-sensor digital camera record only red, green, or blue data. The patterns emphasize the number of green sensors to mimic the human eye's greater sensitivity to green light. The `demosaic` function uses interpolation to convert the two-dimensional Bayer-encoded image into a truecolor image, in the RGB color space.

## **New Function for Creating a Multiresolution Gaussian Pyramid**

The toolbox now supports a function, `impyramid`, that you can use to create a multiresolution Gaussian pyramid. If you specify the `'reduce'` parameter, `impyramid` returns a low-pass filtered version of the image, half the size of the original image. If you specify the `'expand'` parameter, `impyramid` returns a filtered image twice the size of the original image. `impyramid` uses convolution with a Gaussian filter kernel to produce the images.

## **Enhanced ROI Definition Behavior for `imcrop`, `roifill`, and `roipoly`**

**Compatibility Considerations: Yes**

The `imcrop`, `roifill`, and `roipoly` functions now let you define an ROI and then adjust the size and position of the ROI interactively using the mouse. In previous releases, these functions supported the interactive definition of ROIs, but only gave you one chance at the definition. Now, when you are satisfied with the size and shape of the ROI, double-click to perform the cropping, filling, or mask creation operation.

### **Compatibility Considerations**

In previous releases, when defining a polygonal ROI using `roipoly`, pressing **Backspace** deleted the most recent vertex you had defined in the polygon. With this release, pressing **Backspace** deletes the entire polygon. To delete an individual vertex, move the pointer over the vertex, right-click to view the vertex context menu, and then choose **Delete Vertex**.



## **New Modular Interactive GUI-building Tools**

The set of modular interactive tools now includes functions to display a file chooser dialog box and write data to a file (`imsave`). The toolbox also includes a function (`imputfile`) that displays the file chooser dialog box and returns the user's selections.

## New Programmable ROI Tools

The set of programmable ROI creation functions provided by the toolbox now includes three additional shapes:

- Polygons (`impoly`)
- Ellipses (`imellipse`)
- Freehand shapes (`imfreehand`)

The toolbox already includes ROI creation functions to create points, lines, and rectangles.

Each of the ROI creation functions supports an API that you can use to control aspects of its behavior and appearance. For example, you can use API functions to specify the position of the ROI or retrieve the coordinates of its current position.

## Support for Reading NITF and HDR Images

- Read metadata from a National Imagery Transmission Format (NITF) file using `nitfinfo`.
- Read an image from a NITF file using `nitfread`.
- Read high dynamic range (HDR) images using `hdrread`.
- Convert high dynamic range images into a format that can be displayed using the `tonemap` function.

## Enhanced Performance

- Enhanced performance for thinning and skeletonization using `bwmorph`.
- Enhanced performance for filtering RGB images using `imfilter`.

## **DICOM Dictionary Upgrade**

### **Compatibility Considerations: Yes**

The default DICOM dictionary has been upgraded to the 2007 version released by NEMA. A text version of this dictionary is included in the product, `dicom-dict.txt`. This upgrade fixes a problem with the earlier version of the dictionary which contained two instances of the same tag, which caused warnings.

### **Compatibility Considerations**

If your DICOM code depends on hard-coded old attribute names, you may see failures. In addition, some DICOM files may no longer parse. Customers who require attribute settings from the 2005 version can use the `dicomdict` function to access the old data dictionary, which we are shipping in R2007b. That is, `dicom-dict.txt` will have 2007 values and `dicom-dict-2005.txt` is the version of `dicom-dict.txt` found in R2006a and R2007a.

## Changes to Other Functions

This release includes changes to the following functions.

Function	Description of Change
<code>imshow</code>	Is not supported when MATLAB is started with the <code>-nojvm</code> option.
<code>imhist</code>	Can now be embedded in custom GUIs.
<code>fanbeam</code> , <code>ifanbeam</code> , <code>fan2para</code> , <code>para2fan</code>	The fan-beam functions now return different answers than in previous releases due to a bug fix.
<code>imadjdemo</code>	This demo has been deleted from the toolbox.

# R2007a

---

Version: 5.4  
New Features: Yes  
Bug Fixes: Yes

## **Enhancements to `imresize` Function**

### **Compatibility Considerations: Yes**

`imresize` now runs faster, uses less memory, supports new interpolation methods, and supports new options for specifying output size.

### **Compatibility Considerations**

The `imresize` function has been completely rewritten with new algorithms, new options, and new syntaxes. If you need the results produced by the version of `imresize` in previous releases, use the `imresize_old` function.



## **applycform Supports Tetrahedral Interpolation**

**Compatibility Considerations: Yes**

The applycform function now uses tetrahedral interpolation for profiles containing multidimensional lookup tables, and returns more accurate results.

### **Compatibility Considerations**

The results returned by applycform are more accurate but they are different than results returned in previous releases, for profiles containing multidimensional lookup tables.

## **Control Point Selection Tool Enhancements**

### **Compatibility Considerations: Yes**

The Control Point Selection Tool has enhanced visual appearance and usability. For example, points are now numbered for easier identification of matched pairs.

In addition, the tool now supports a 'wait' option which enables `cpselect` to be used in scripts. When you specify this option, `cpselect` blocks the MATLAB command line until point selection is completed. For information about using the Control Point Selection Tool, see Image Registration and the reference page for the `cpselect` function.

### **Compatibility Considerations**

The Control Point Selection Tool no longer includes the **Redo** or **Undo** options on the Edit menu.

## Enhancements to `impoint`, `imline`, and `imrect` Functions

The `impoint`, `imline`, and `imrect` function now support an interactive placement capability. Using the mouse, you can specify the initial position of the point, line, or rectangle. In addition, the `imrect` function now supports interactive resizing using the mouse. See the reference pages for these functions for more information and examples.

## **Enhancements to montage Function**

The `montage` function now supports parameters that control the arrangement and appearance of the images displayed. See the `montage` reference page for these functions for more information and examples

---

## Compatibility Considerations: Yes

### Compatibility Considerations

#### **makecform Uses 'icc' Whitepoint for L\*a\*b\*/sRGB Conversions**

The `makecform` function now only uses the white point type 'icc' for color space conversions from  $L^*a^*b^*$  to *srgb* (type = 'lab2srgb') and from *srgb* to  $L^*a^*b^*$  (type = 'srgb2lab'). In previous releases, you could specify other white point values for these conversions, using the optional 'Whitepoint' parameter. This syntax now issues a warning when any other white point besides 'icc' is specified.

#### **normxcorr2 Might Return Different Results**

The `normxcorr2` function now returns values in the range [-1,1] for all inputs. In previous releases, `normxcorr2` returned values outside this range for certain inputs.

#### **watershed Uses New Algorithm**

The watershed transform algorithm used by the `watershed` function has changed. The previous algorithm occasionally produced labeled watershed basins that were not contiguous. If you need to obtain the same results as the previous algorithm, use the function `watershed_old`.

## Changes to Other Functions

This release includes changes to the following functions.

Function	Description of Change
<code>imshow</code>	New 'border' parameter, to control whether <code>imshow</code> includes a border around the image displayed, and 'parent' parameter, to specify the axes in which to display the image.
<code>imshowpanel</code>	New 'replaceImage' parameter lets you replace the image displayed in the scroll panel with a new image.
<code>iradon</code>	New 'none' value for the <code>filter</code> parameter returns an unfiltered backprojection; also supports new interpolation types.
<code>iptsetpref</code>	New <code>UseIPPL</code> preference.

# R2006b

---

Version: 5.3  
New Features: Yes  
Bug Fixes: Yes

## Enhancements to DICOM Capabilities

This release includes the following new features and enhancements to the DICOM capabilities of the Image Processing Toolbox:

- The toolbox includes a new function, `dicomlookup`, that provides a way to find the name of an attribute in a DICOM data dictionary by specifying its group and element tags, or find the group and element tags for an attribute by specifying its name.
- The performance of the `dicominfo` function has been significantly improved



## **New Symmetric Option with `graycomatrix` Function**

The `graycomatrix` function now supports a new option: `'symmetric'`. With this option, you can create a gray-level co-occurrence matrix (GLCM) that is symmetric about its diagonal. This is consistent with the GLCM definition given by Haralick in his 1973 article. For more information, see `graycomatrix`.

## **Enhancements to ICC Color Capabilities**

### **Compatibility Considerations: Yes**

The toolbox includes the following enhancements to the ICC color capabilities:

- The `applycform` function can now transform colors using profiles that contain parametric curve types.
- The `iccread` function now supports named colors in ICC profiles.

### **Compatibility Considerations**

The `whitepoint` function, when used with the `'d50'` argument, returns different results in R2006b than it did in R2006a. The previously returned *XYZ* color values were incorrect according to the current interpretation of standards. If your algorithm depended on the old values, you might see subtly different results.

## Enhancements to the imdistline Function

### Compatibility Considerations: Yes

This release includes the following enhancements to the `imdistline` function:

- The `imdistline` function now uses a different cursor shape at its endpoints to highlight that these endpoints can be grabbed to change the length or direction of the line. The function uses a hand cursor over endpoints and a fleur cursor over the body of the line.
- The `imdistline` function reference page now includes an example that shows how to use the `XData` and `YData` properties of the associated image to express distance in non-pixel units.

### Compatibility Considerations

The Distance Tool's `getAngleFromHorizontal` method now returns a value between 0 and 180 degrees. Previously, this function incorrectly returned a value between 0 and 90. For an explanation of how `getAngleFromHorizontal` calculates this angle, see the `imdistline` function.

## **setColor Method Accepts Predefined Color Strings**

The `setColor` method of the `imshow`, `imline`, `impoint`, and `imrect` functions accepts an RGB triplet or the short- or long-name version of the MATLAB predefined color names.

# R2006a

---

Version: 5.2  
New Features: Yes  
Bug Fixes: Yes

## Enhanced ICC Profile Capabilities

The `iccread` and `iccwrite` functions have been updated to support recent changes to the ICC specification.

In addition, `iccread` can now read and process the following additional profile types:

- DeviceLink profiles — Provide transformation from one device space to another.
- ColorSpace profiles — Provide transformation between a non-device color space and the profile connection space (PCS).
- Abstract profiles — Enable color transformations to be defined that provide specific color effects.
- Grayscale profiles — Specify the relationship between device values and the PCS for specific colors.

In addition, `iccread` can now read parametric curve types.

## **New Pointer Management Functions**

The toolbox includes three new utility functions, `iptPointerManager`, `iptGetPointerBehavior`, and `iptSetPointerBehavior`, that you can use to manage changes to the pointer in GUIs. For example, you can use the pointer management functions to change the appearance of the pointer when it moves over objects in a figure. These functions can be useful when building GUIs with the toolbox modular GUI tools.

## **New Constraint Creation Function**

The toolbox includes a new utility function, `makeConstrainToRectFcn`, that you can use to specify drag constraints for the `imdistline`, `imline`, `impoint`, and `imrect` functions. You specify the constraints as arguments to the `makeConstrainToRectFcn` and this function returns a handle to a constraint function. To use this constraint with an object, set the value of the `setConstraintFcn` API for the object to this function handle.



## Functions `cp2tform`, `tforminv`, `imtransform`

**Compatibility Considerations: Yes**

### Compatibility Considerations

When using the `cp2tform`, `tforminv`, or `imtransform` functions with the transform type `'piecewise linear'` you might get different answers from previous versions due to a bug fix. If you have a transformation structure (TFORM) saved from an older version, you may want to regenerate it from control points to get improved performance.

## **IPPL Not Used on 64-Bit Systems**

Certain functions in the Image Processing Toolbox, such as the image arithmetic functions, use the Intel Performance Primitives Library (IPPL), if it's available. (See `ipp1` for more information.) Note that these functions do not use the IPPL on 64-bit systems.

# R14SP3

---

Version: 5.1  
New Features: Yes  
Bug Fixes: Yes

## **Support for Two New Medical Image File Formats**

The toolbox now includes functions for reading metadata and image data from two additional medical image file formats. Analyze 7.5 and Interfile. For more information, see Reading and Writing Data in Medical Formats.

## **New Point, Rectangle, and Line Functions**

The toolbox includes three functions, `impoint`, `imline`, and `imrect`, that you can use to create draggable points, lines, and rectangles in a figure window. These functions can be used as building blocks for other GUI tools.

## **Image Tool Enhancements and Improvements**

### **New Distance Tool**

The Image Tool now includes a new Distance tool that you can use to determine the distance between any two points in an image. This tool is also available in the toolbox's suite of modular interactive GUI tools. Using the `imdistanline` function you can add the Distance tool to GUIs of your own creation. For more information, see [Measuring Features in an Image](#)

### **Adjust Contrast Tool Enhancements and Improvements**

The Adjust Contrast tool has been redesigned to provide better usability. For examples, the Adjust Contrast tool Window/Level capability is now a separate mode with its own activation button.

## **New Utility Functions for Use with Profile-Based Color Space Conversion Functions**

The toolbox has two new utility functions, `iccroot` and `iccfind`, for use with the profile-based color conversion functions. For more information, see [Performing Profile-based Color Space Conversions](#).

## **New Documentation on Processing Image Sequences**

The Image Processing Toolbox User's Guide includes a new section, Working with Image Sequences, that describes which toolbox functions can be used with sequences of image, also known as image stacks



## **Control Point Selection Tool Now Works on Macintosh Systems**

The Control Point Selection Tool now works on Macintosh systems.

## Obsolete and Deleted Functions

Compatibility Considerations: Yes

### Compatibility Considerations

The following table lists toolbox functions that have been made obsolete or removed in this version.

Function	Enhancement
<code>impositionrect</code>	This function is obsolete. Use <code>imrect</code> to perform the same tasks.
<code>pixval</code>	This function is obsolete. It now issues a warning when used. Use <code>impixelinfo</code> for pixel reporting and use <code>imdistline</code> for measuring distance

## **Image Tool is Not Compilable**

### **Compatibility Considerations**

The `imtool` function is not compilable with the MATLAB Compiler.



# R14SP2

---

Version: 5.0.2  
New Features: Yes  
Bug Fixes: No

## Major Bug Fixes

This release contains the following bug fixes.

## Major Revisions to Fan-Beam Functions

This release includes numerous updates and improvements to the fan-beam functions: `fanbeam`, `ifanbeam`, `fan2para`, and `para2fan`. The fixes include improved calculations, improved documentation, and examples.

For example, `fanbeam` now returns the correct sensor locations when the geometry is `'line'`. The `ifanbeam` and `fan2para` now consistently use the correct default value for the `'FanSensorSpacing'` parameter. If you tried the fan-beam functions in a previous release, you might try them again to take advantage of these improvements.

In addition to the functional changes, many improvements to the documentation of the fan-beam functions have been made.

`fanbeam` help now includes

- An example that shows how to extract projection data at a specific rotation angle from the fan-beam data returned
- An explanation of how `fanbeam` calculates the number of rows and columns in `F`, the fan-beam data returned
- The default value for the `'FanSensorSpacing'` parameter for both `'line'` and `'arc'` geometries
- Guidelines for setting the value of the `D` parameter

The help for the `ifanbeam` function now includes an example that shows how to use the `'minimal'` coverage parameter.

## Compatibility Considerations

Results computed with earlier versions of the fan-beam functions cannot be used with the new versions of these functions.

## Changes to the DICOM Functions

The following fixes have been made to the `dicomread` and `dicomwrite` functions.

Function	Bug Fixes
dicomread	No longer errors when reading files that contain extraneous pixel data; instead, <code>dicomread</code> issues a warning message. However, if the file does not contain enough pixel data, <code>dicomread</code> issues an error.
dicomwrite	<ul style="list-style-type: none"> <li>• No longer is case sensitive when parsing input parameters. For example, you can specify either <code>'CreateMode'</code> or <code>'createmode'</code>.</li> <li>• Preserves the full precision of data converted to decimal string metadata. Previously, <code>dicomwrite</code> limited precision to six digits.</li> <li>• No longer errors when writing files with metadata values that must be stored as a decimal string or integer string. Now, when writing private data attributes (attributes that are not listed in the DICOM data dictionary), <code>dicomwrite</code> assigns the attributes the type UN (for unknown) and writes the data to the file as a byte-for-byte copy of its in-memory representation. Because <code>dicomwrite</code> writes the file with explicit value representation (VR), the file might have a different VR value, but the data will be the same.</li> <li>• Includes the <code>TriggerTime</code> field for additional values of <code>ScanOptions</code>, including <code>'CT'</code>. Previously, <code>dicomwrite</code> only included the <code>TriggerTime</code> attribute if the <code>ScanOptions</code> field indicated a gated heart MR.</li> <li>• No longer issues an <code>Unsupported SOP class</code> error message if, when <code>'create'</code> mode is specified, semantic verification is not available for an information object. Instead, <code>dicomwrite</code> issues a more helpful message indicating that it might be able to write the data if the mode was <code>'copy'</code>, rather than <code>'create'</code>. In <code>'copy'</code> mode, <code>dicomwrite</code> only performs syntactic checking, not semantic verification. Consequently, <code>dicomwrite</code> can write many more types of DICOM files in <code>'copy'</code> mode than it can in <code>'create'</code> mode. See the <code>dicomwrite</code> reference page for important information about data integrity.</li> </ul>

## Changes to Image Tool and Modular Interactive Tools

The following fixes have been made to the Image Tool and other modular interactive tools:

- The Image Tool now always makes the **Open** and **Import from Workspace** options available on its **File** menu. Previously, the Image Tool disabled these options if the tool contained an image. If the Image Tool contains an image, the newly imported image is displayed in a new Image Tool using the default preferences.
- The Image Tool zoom buttons can now be used on an image that has superimposed vector data.
- The Image Tool toolbar buttons no longer create multiple versions of the modular interactive tools when clicked rapidly in quick succession.
- The Image Information tool now displays correctly on Linux systems. Previously, it displayed as a blank window.
- The Overview tool can now be resized from any corner. Previously, resizing the tool using a corner other than the lower left caused the image to become progressively smaller until it disappeared.
- The Overview tool zoom buttons now provide an affordance that informs users when they cannot use these buttons to zoom in or out on the image displayed in the associated scroll panel.
- The Pixel Region tool now displays floating-point values correctly. Previously, the pixel value text strings displayed spilled over into adjacent pixels for some floating-point images.
- The Pixel Region tool now works correctly with images displayed in subplots.
- The Pixel Region tool no longer causes the target image to become tiny and move to a different position in the figure.

## Changes to the imshow Function

- The `imshow` function no longer overwrites nondefault axes in a figure.
- The `imshow` function ignores any initial magnification value you specify when used to display an image in a figure that is docked (the figure's `WindowState` property is set to `'docked'`). In these cases, `imshow` displays



the image at the largest magnification that fits the window ('fit' magnification) and issues a warning.

## Fixes to Other Functions

The following tables lists fixes that have been done to other toolbox functions.

Function	Enhancement
<code>applycform</code>	Now correctly handles profiles that contain a gamut tag.
<code>cpcorr</code>	Now is more numerically robust. For this release, the subfunction <code>findpeak</code> , which <code>cpcorr</code> calls, has been improved and is now a private function, rather than a subfunction.
<code>imhist</code>	No longer causes a docked figure window to become undocked.
<code>imrotate</code>	Now correctly rotates N-dimensional arrays, where N is greater than 3. In previous releases, <code>imrotate</code> would accept N-D arrays but only return a 3-D array.
<code>normxcorr2</code>	Now always returns real values. In previous releases, due to roundoff error, some sets of input data caused the <code>normxcorr2</code> function to return a complex valued matrix of correlation coefficients.
<code>pixval</code>	Now works correctly with binary images.
<code>rgb2ind</code>	Now returns a correct output image when called with the syntax <code>rgb2ind(rgb,n,'nodither')</code> where n is greater than 256.

## Fixes to Image Processing Toolbox Deployment Issues

- Performance issues that occurred when deploying compiled image processing toolbox functions that call IPPLE routines have been fixed.

- Running compiled versions of `imtool` and some of the other modular interactive tools no longer generates the following warning messages about classes not being cleared:

Warning: Objects of `graphics.linkprop` class exist - not clearing this class or any of its super-classes.

Warning: An object instance still exists. Use the `objectdirectory` command to see a count of existing instances.

## **New Directory Needed**

**Compatibility Considerations: Yes**

### **Compatibility Considerations**

The Image Processing Toolbox software now requires the following new directory on the MATLAB path:

```
toolbox\shared\imageslib
```