

Display Images with Image Viewer App

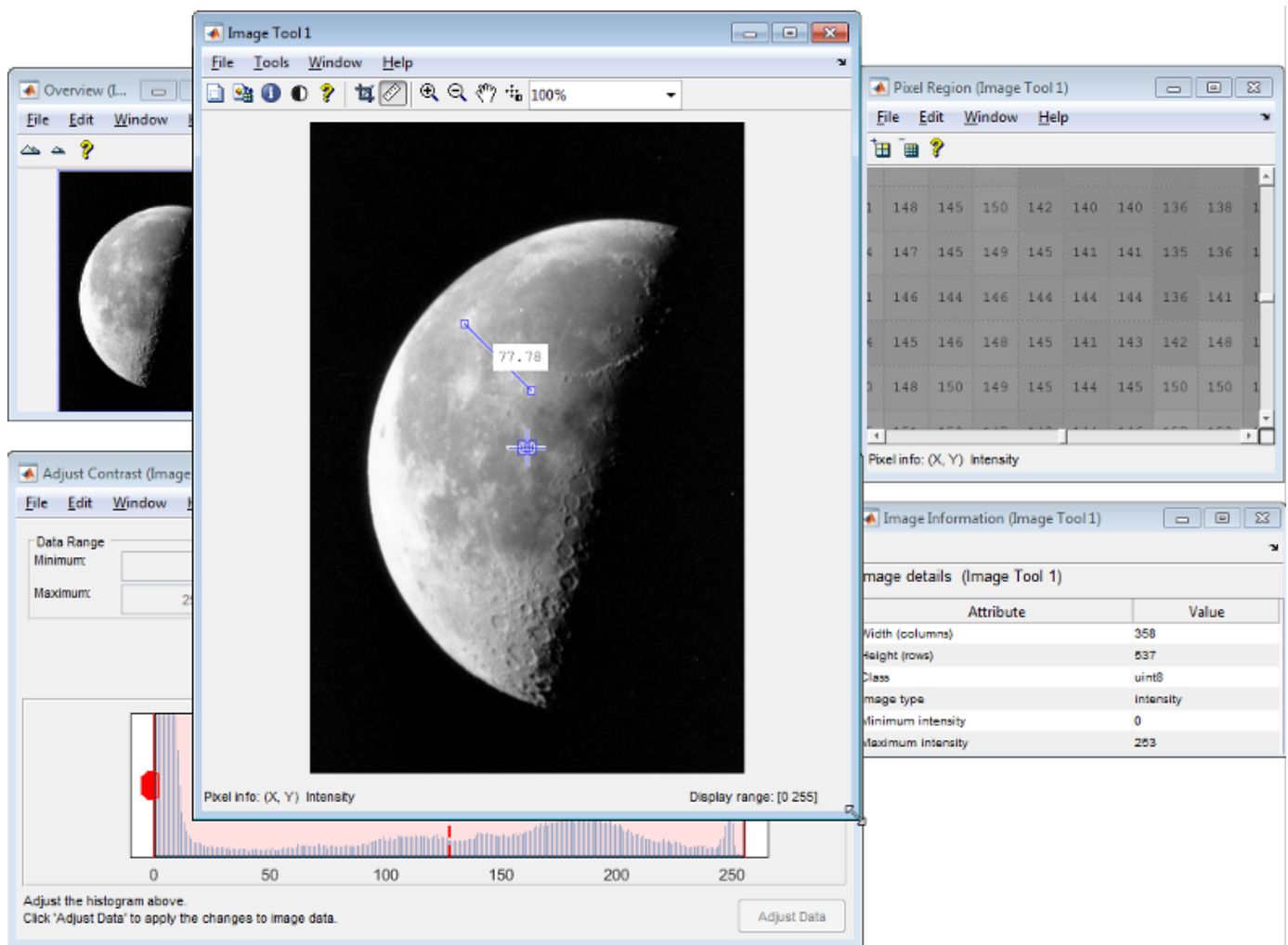
The Image Viewer app is an image display and exploration tool that presents an integrated environment for displaying images and performing common image processing tasks. The Image Viewer provides access to several other tools:

- Pixel Information tool — for getting information about the pixel under the pointer
- Pixel Region tool — for getting information about a group of pixels
- Distance tool — for measuring the distance between two pixels
- Image Information tool — for getting information about image and image file metadata
- Adjust Contrast tool and associated Window/Level tool — for adjusting the contrast of the image displayed in the Image Viewer and modifying the actual image data. You can save the adjusted data to the workspace or a file.
- Crop Image tool — for defining a crop region on the image and cropping the image. You can save the cropped image to the workspace or a file.
- Display Range tool — for determining the display range of the image data

In addition, the Image Viewer provides several navigation aids that can help explore large images:

- Overview tool — for determining what part of the image is currently visible in the Image Viewer and changing this view.
- Pan tool — for moving the image to view other parts of the image
- Zoom tool — for getting a closer view of any part of the image.
- Scroll bars — for navigating over the image.

The following figure shows the image displayed in the Image Viewer app with many of the related tools open and active.



Open the Image Viewer App

To start the Image Viewer, click **Image Viewer** on the **Apps** tab, or use the `imtool` function. You can also start another Image Viewer from within an existing Image Viewer by using the **New** option from the **File** menu.

To bring image data into the Image Viewer, you can use either the **Open** or **Import from Workspace** options from the **File** menu — see [Import Image Data from the Workspace into the Image Viewer App](#).

You can also specify the name of the MATLAB[®] workspace variable that contains image data when you call `imtool`, as follows:

```
moon = imread('moon.tif');
imtool(moon)
```

Alternatively, you can specify the name of the graphics file containing the image. This syntax can be useful for scanning through graphics files.

```
imtool('moon.tif');
```

Note When you specify a file name, the image data is not stored in a MATLAB workspace variable. To bring the image displayed in the Image Viewer into the workspace, you must use the `getimage` function or the **Export to Workspace** option from the Image Viewer **File** menu — see [Export Image Data from the Image Viewer App to the Workspace](#).

Initial Image Magnification in the Image Viewer App

The Image Viewer attempts to display an image in its entirety at 100% magnification (one screen pixel for each image pixel) and always honors any magnification value you specify. If the image is too big to fit in a figure on the screen, the Image Viewer shows only a portion of the image, adding scroll bars to allow navigation to parts of the image that are not currently visible. If the specified magnification would make the image too large to fit on the screen, the Image Viewer scales the image to fit, without issuing a warning. This is the default behavior, specified by the 'InitialMagnification' parameter value 'adaptive'.

To override this default initial magnification behavior for a particular call to `imtool`, specify the `InitialMagnification` parameter. For example, to view an image at 150% magnification, use this code.

```
pout = imread('pout.tif');  
imtool(pout, 'InitialMagnification', 150)
```

You can also specify the `'fit'` as the initial magnification value. In this case, `imtool` scales the image to fit the default size of a figure window.

Another way to change the default initial magnification behavior of the Image Viewer is to set the `ImtoolInitialMagnification` toolbox preference. The magnification value you specify remains in effect until you change it. To set the preference, use `iptsetpref` or open the Image Processing Toolbox™ Preferences panel. To open the preferences panel, call `iptprefs` or select **File > Preferences** in the Image Viewer menu. To learn more about toolbox preferences, see [iptprefs](#).

When the Image Viewer scales an image, it uses interpolation to determine the values for screen pixels that do not directly correspond to elements in the image matrix. For more information about specifying an interpolation method, see [Resize an Image with imresize Function](#).

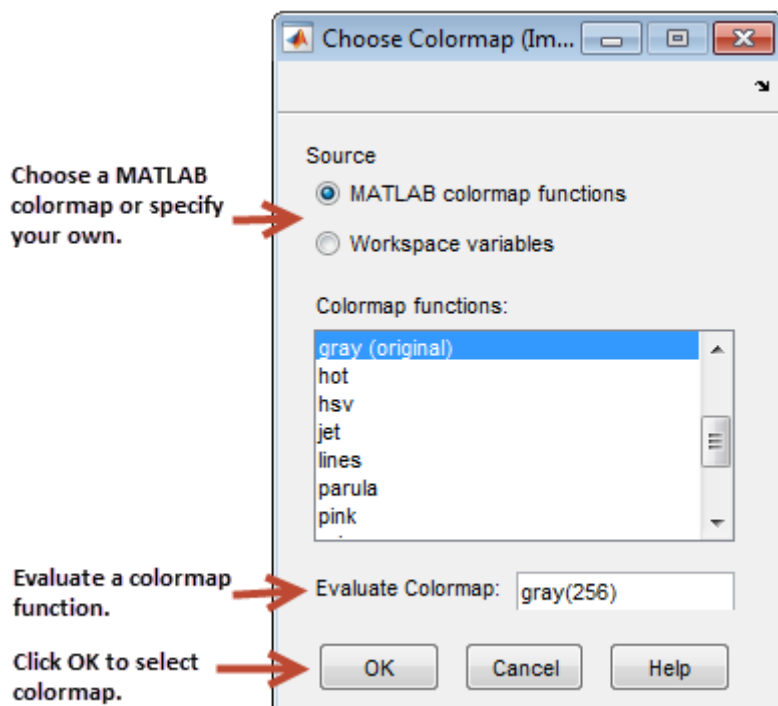
Choose the Colormap Used by the Image Viewer App

A colormap is a matrix that can have any number of rows, but must have three columns. Each row in the colormap is interpreted as a color, with the first element specifying the intensity of red, the second green, and the third blue.

To specify the color map used to display an indexed image or a grayscale image in the Image Viewer, select the **Choose Colormap** option on the **Tools** menu. This activates the Choose Colormap tool. Using this tool you can select one of the MATLAB colormaps or select a colormap variable from the MATLAB workspace.

When you select a colormap, the Image Viewer executes the colormap function you specify and updates the image displayed. You can edit the colormap command in the **Evaluate Colormap** text box; for example, you can change the number of entries in the colormap (default is 256). You can enter your own colormap function in this field. Press **Enter** to execute the command.

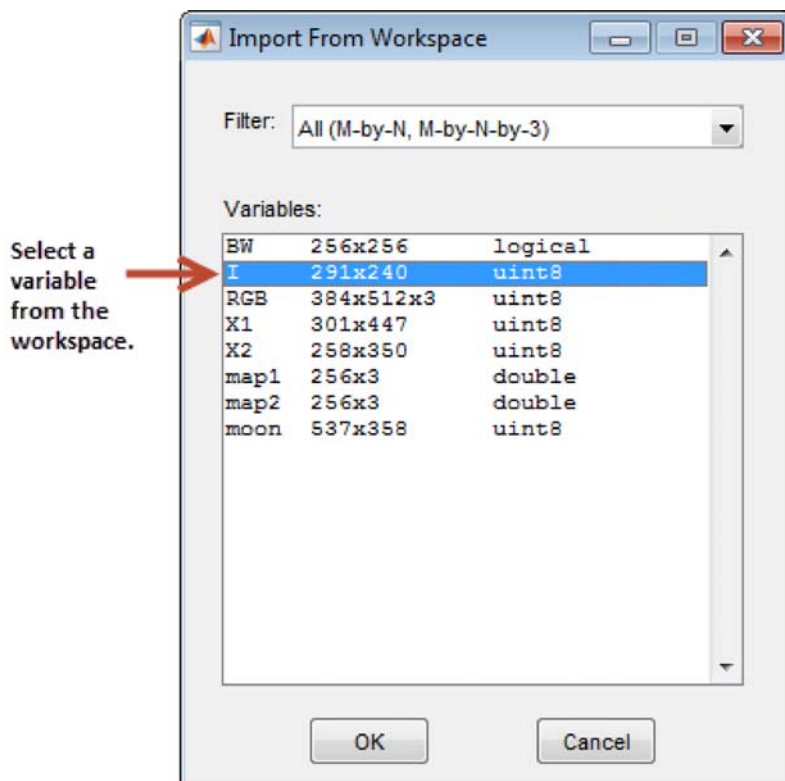
When you choose a colormap, the image updates to use the new map. If you click **OK**, the Image Viewer applies the colormap and closes the Choose Colormap tool. If you click **Cancel**, the image reverts to the previous colormap.



Import Image Data from the Workspace into the Image Viewer App

To import image data from the MATLAB workspace into the Image Viewer, use the **Import from Workspace** option on the Image Viewer **File** menu. In the dialog box, shown below, you select the workspace variable that you want to import into the workspace.

The following figure shows the Import from Workspace dialog box. You can use the Filter menu to limit the images included in the list to certain image types, i.e., binary, indexed, intensity (grayscale), or truecolor.



Export Image Data from the Image Viewer App to the Workspace

To export the image displayed in the Image Viewer to the MATLAB workspace, you can use the **Export to Workspace** option on the Image Viewer **File** menu. (Note that when exporting data, changes to the display range will not be preserved.) In the dialog box, shown below, you specify the name you want to assign to the variable in the workspace. By default, the Image Viewer prefills the variable name field with BW, for binary images, RGB, for truecolor images, and I for grayscale or indexed images.

If the Image Viewer contains an indexed image, this dialog box also contains a field where you can specify the name of the associated colormap.



Using the getimage Function to Export Image Data

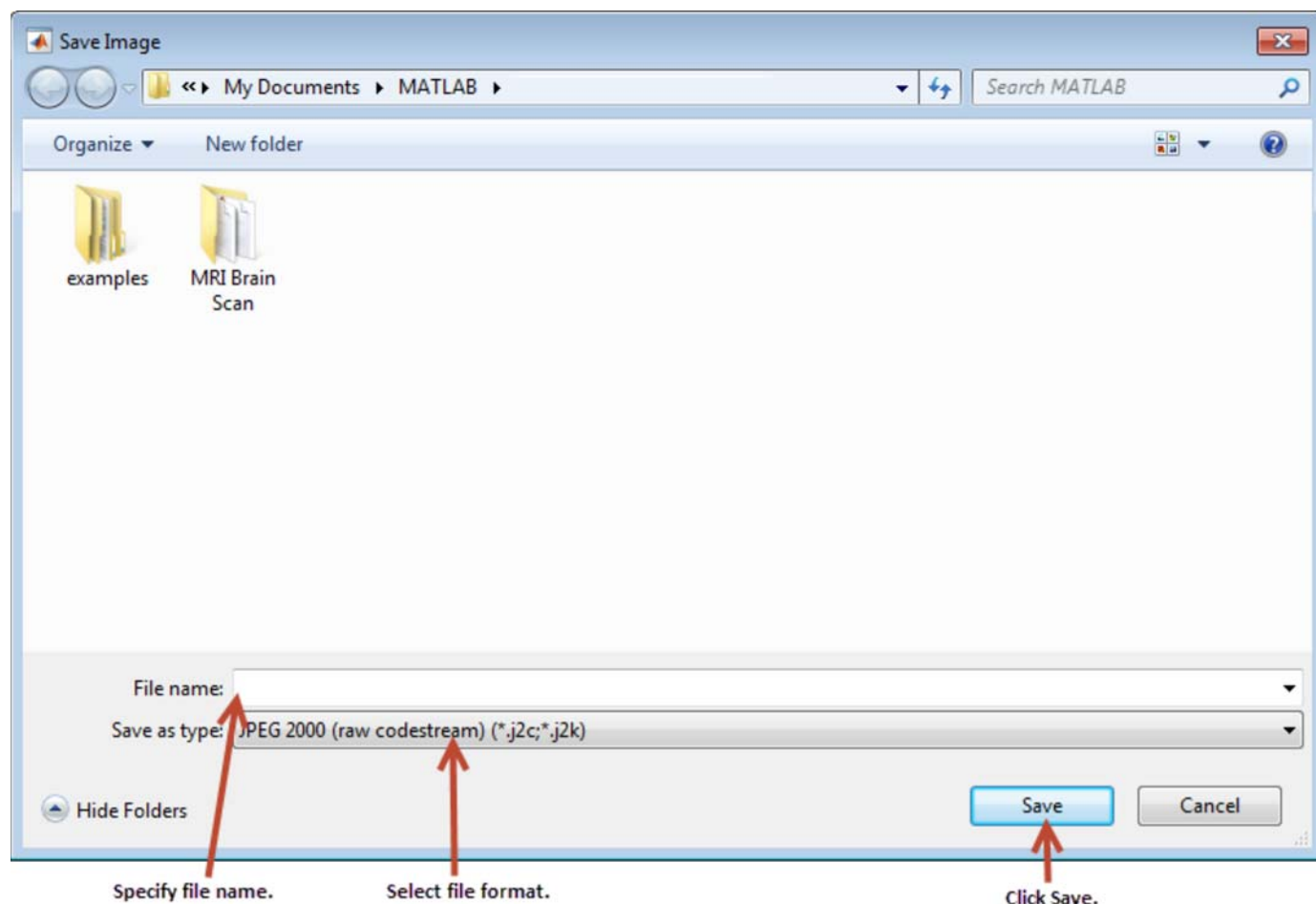
You can also use the `getimage` function to bring image data from the Image Viewer into the MATLAB workspace.

The `getimage` function retrieves the image data (CData) from the current image object. You must use the toolbox function `imgca` to get the image object displayed in the Image Viewer. The following example assigns the image data from `moon.tif` to the variable `moon` if the figure window in which it is displayed is currently active.

```
moon = getimage(imgca);
```

Save Image Data Displayed in Image Viewer

To save the image data displayed in the Image Viewer, select the **Save as** option from the Image Viewer **File** menu. The Image Viewer opens the Save Image dialog box, shown in the following figure. Use this dialog box to navigate your file system to determine where to save the image file and specify the name of the file. Choose the graphics file format you want to use from among many common image file formats listed in the Files of Type menu. If you do not specify a file name extension, the Image Viewer adds an extension to the file associated with the file format selected, such as .jpg for the JPEG format.



Note: Changes you make to the display range will not be saved. If you would like to preserve your changes, use [imcontrast](#).

Close the Image Viewer App

To close the Image Viewer, use the **Close** button in the window title bar or select the **Close** option from the Image Viewer **File** menu. If you used the `imtool` function to start the Image Viewer you can get the figure object that contains the app. You can use this object to close the app. When you close the Image Viewer, any related tools that are currently open also close.

Because the Image Viewer does not make its component graphics objects visible, the Image Viewer does not close when you call the MATLAB `close all` command. If you want to close multiple Image Viewers, use the syntax

```
imtool close all
```

or select **Close all** from the Image Viewer **File** menu.

Print Images Displayed in Image Viewer App

To print the image displayed in the Image Viewer, select the **Print to Figure** option from the **File** menu. The Image Viewer opens another figure window and displays the image. Use the **Print** option on the **File** menu of this figure window to print the image. See [Print Images](#) for more information.