| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ImageObserver.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/image/ImageFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/ImageProducer.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/ImageObserver.html)    [**NO FRAMES**](http://docs.google.com/ImageObserver.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#35nkun2) |

## **java.awt.image**

Interface ImageObserver

**All Known Implementing Classes:** [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html), [AbstractColorChooserPanel](http://docs.google.com/javax/swing/colorchooser/AbstractColorChooserPanel.html), [Applet](http://docs.google.com/java/applet/Applet.html), [BasicArrowButton](http://docs.google.com/javax/swing/plaf/basic/BasicArrowButton.html), [BasicComboBoxRenderer](http://docs.google.com/javax/swing/plaf/basic/BasicComboBoxRenderer.html), [BasicComboBoxRenderer.UIResource](http://docs.google.com/javax/swing/plaf/basic/BasicComboBoxRenderer.UIResource.html), [BasicComboPopup](http://docs.google.com/javax/swing/plaf/basic/BasicComboPopup.html), [BasicInternalFrameTitlePane](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.html), [BasicInternalFrameTitlePane.SystemMenuBar](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.SystemMenuBar.html), [BasicSplitPaneDivider](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html), [BasicToolBarUI.DragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DragWindow.html), [Box](http://docs.google.com/javax/swing/Box.html), [Box.Filler](http://docs.google.com/javax/swing/Box.Filler.html), [Button](http://docs.google.com/java/awt/Button.html), [Canvas](http://docs.google.com/java/awt/Canvas.html), [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html), [Checkbox](http://docs.google.com/java/awt/Checkbox.html), [Choice](http://docs.google.com/java/awt/Choice.html), [Component](http://docs.google.com/java/awt/Component.html), [Container](http://docs.google.com/java/awt/Container.html), [DefaultListCellRenderer](http://docs.google.com/javax/swing/DefaultListCellRenderer.html), [DefaultListCellRenderer.UIResource](http://docs.google.com/javax/swing/DefaultListCellRenderer.UIResource.html), [DefaultTableCellRenderer](http://docs.google.com/javax/swing/table/DefaultTableCellRenderer.html), [DefaultTableCellRenderer.UIResource](http://docs.google.com/javax/swing/table/DefaultTableCellRenderer.UIResource.html), [DefaultTreeCellEditor.DefaultTextField](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.DefaultTextField.html), [DefaultTreeCellEditor.EditorContainer](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.EditorContainer.html), [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html), [Dialog](http://docs.google.com/java/awt/Dialog.html), [FileDialog](http://docs.google.com/java/awt/FileDialog.html), [Frame](http://docs.google.com/java/awt/Frame.html), [JApplet](http://docs.google.com/javax/swing/JApplet.html), [JButton](http://docs.google.com/javax/swing/JButton.html), [JCheckBox](http://docs.google.com/javax/swing/JCheckBox.html), [JCheckBoxMenuItem](http://docs.google.com/javax/swing/JCheckBoxMenuItem.html), [JColorChooser](http://docs.google.com/javax/swing/JColorChooser.html), [JComboBox](http://docs.google.com/javax/swing/JComboBox.html), [JComponent](http://docs.google.com/javax/swing/JComponent.html), [JDesktopPane](http://docs.google.com/javax/swing/JDesktopPane.html), [JDialog](http://docs.google.com/javax/swing/JDialog.html), [JEditorPane](http://docs.google.com/javax/swing/JEditorPane.html), [JFileChooser](http://docs.google.com/javax/swing/JFileChooser.html), [JFormattedTextField](http://docs.google.com/javax/swing/JFormattedTextField.html), [JFrame](http://docs.google.com/javax/swing/JFrame.html), [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html), [JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html), [JLabel](http://docs.google.com/javax/swing/JLabel.html), [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html), [JList](http://docs.google.com/javax/swing/JList.html), [JMenu](http://docs.google.com/javax/swing/JMenu.html), [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html), [JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html), [JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html), [JPanel](http://docs.google.com/javax/swing/JPanel.html), [JPasswordField](http://docs.google.com/javax/swing/JPasswordField.html), [JPopupMenu](http://docs.google.com/javax/swing/JPopupMenu.html), [JPopupMenu.Separator](http://docs.google.com/javax/swing/JPopupMenu.Separator.html), [JProgressBar](http://docs.google.com/javax/swing/JProgressBar.html), [JRadioButton](http://docs.google.com/javax/swing/JRadioButton.html), [JRadioButtonMenuItem](http://docs.google.com/javax/swing/JRadioButtonMenuItem.html), [JRootPane](http://docs.google.com/javax/swing/JRootPane.html), [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html), [JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html), [JScrollPane.ScrollBar](http://docs.google.com/javax/swing/JScrollPane.ScrollBar.html), [JSeparator](http://docs.google.com/javax/swing/JSeparator.html), [JSlider](http://docs.google.com/javax/swing/JSlider.html), [JSpinner](http://docs.google.com/javax/swing/JSpinner.html), [JSpinner.DateEditor](http://docs.google.com/javax/swing/JSpinner.DateEditor.html), [JSpinner.DefaultEditor](http://docs.google.com/javax/swing/JSpinner.DefaultEditor.html), [JSpinner.ListEditor](http://docs.google.com/javax/swing/JSpinner.ListEditor.html), [JSpinner.NumberEditor](http://docs.google.com/javax/swing/JSpinner.NumberEditor.html), [JSplitPane](http://docs.google.com/javax/swing/JSplitPane.html), [JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html), [JTable](http://docs.google.com/javax/swing/JTable.html), [JTableHeader](http://docs.google.com/javax/swing/table/JTableHeader.html), [JTextArea](http://docs.google.com/javax/swing/JTextArea.html), [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html), [JTextField](http://docs.google.com/javax/swing/JTextField.html), [JTextPane](http://docs.google.com/javax/swing/JTextPane.html), [JToggleButton](http://docs.google.com/javax/swing/JToggleButton.html), [JToolBar](http://docs.google.com/javax/swing/JToolBar.html), [JToolBar.Separator](http://docs.google.com/javax/swing/JToolBar.Separator.html), [JToolTip](http://docs.google.com/javax/swing/JToolTip.html), [JTree](http://docs.google.com/javax/swing/JTree.html), [JViewport](http://docs.google.com/javax/swing/JViewport.html), [JWindow](http://docs.google.com/javax/swing/JWindow.html), [Label](http://docs.google.com/java/awt/Label.html), [List](http://docs.google.com/java/awt/List.html), [MetalComboBoxButton](http://docs.google.com/javax/swing/plaf/metal/MetalComboBoxButton.html), [MetalComboBoxUI.MetalComboPopup](http://docs.google.com/javax/swing/plaf/metal/MetalComboBoxUI.MetalComboPopup.html), [MetalFileChooserUI.FileRenderer](http://docs.google.com/javax/swing/plaf/metal/MetalFileChooserUI.FileRenderer.html), [MetalFileChooserUI.FilterComboBoxRenderer](http://docs.google.com/javax/swing/plaf/metal/MetalFileChooserUI.FilterComboBoxRenderer.html), [MetalInternalFrameTitlePane](http://docs.google.com/javax/swing/plaf/metal/MetalInternalFrameTitlePane.html), [MetalScrollButton](http://docs.google.com/javax/swing/plaf/metal/MetalScrollButton.html), [Panel](http://docs.google.com/java/awt/Panel.html), [Scrollbar](http://docs.google.com/java/awt/Scrollbar.html), [ScrollPane](http://docs.google.com/java/awt/ScrollPane.html), [TextArea](http://docs.google.com/java/awt/TextArea.html), [TextComponent](http://docs.google.com/java/awt/TextComponent.html), [TextField](http://docs.google.com/java/awt/TextField.html), [Window](http://docs.google.com/java/awt/Window.html)

public interface **ImageObserver**

An asynchronous update interface for receiving notifications about Image information as the Image is constructed.

| **Field Summary** | |
| --- | --- |
| static int | [**ABORT**](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT)            This flag in the infoflags argument to imageUpdate indicates that an image which was being tracked asynchronously was aborted before production was complete. |
| static int | [**ALLBITS**](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS)            This flag in the infoflags argument to imageUpdate indicates that a static image which was previously drawn is now complete and can be drawn again in its final form. |
| static int | [**ERROR**](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR)            This flag in the infoflags argument to imageUpdate indicates that an image which was being tracked asynchronously has encountered an error. |
| static int | [**FRAMEBITS**](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS)            This flag in the infoflags argument to imageUpdate indicates that another complete frame of a multi-frame image which was previously drawn is now available to be drawn again. |
| static int | [**HEIGHT**](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT)            This flag in the infoflags argument to imageUpdate indicates that the height of the base image is now available and can be taken from the height argument to the imageUpdate callback method. |
| static int | [**PROPERTIES**](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES)            This flag in the infoflags argument to imageUpdate indicates that the properties of the image are now available. |
| static int | [**SOMEBITS**](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS)            This flag in the infoflags argument to imageUpdate indicates that more pixels needed for drawing a scaled variation of the image are available. |
| static int | [**WIDTH**](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH)            This flag in the infoflags argument to imageUpdate indicates that the width of the base image is now available and can be taken from the width argument to the imageUpdate callback method. |

| **Method Summary** | |
| --- | --- |
| boolean | [**imageUpdate**](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))([Image](http://docs.google.com/java/awt/Image.html) img, int infoflags, int x, int y, int width, int height)            This method is called when information about an image which was previously requested using an asynchronous interface becomes available. |

| **Field Detail** |
| --- |

### WIDTH

static final int **WIDTH**

This flag in the infoflags argument to imageUpdate indicates that the width of the base image is now available and can be taken from the width argument to the imageUpdate callback method.

**See Also:**[Image.getWidth(java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Image.html#getWidth(java.awt.image.ImageObserver)), [imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.WIDTH)

### HEIGHT

static final int **HEIGHT**

This flag in the infoflags argument to imageUpdate indicates that the height of the base image is now available and can be taken from the height argument to the imageUpdate callback method.

**See Also:**[Image.getHeight(java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Image.html#getHeight(java.awt.image.ImageObserver)), [imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.HEIGHT)

### PROPERTIES

static final int **PROPERTIES**

This flag in the infoflags argument to imageUpdate indicates that the properties of the image are now available.

**See Also:**[Image.getProperty(java.lang.String, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Image.html#getProperty(java.lang.String,%20java.awt.image.ImageObserver)), [imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.PROPERTIES)

### SOMEBITS

static final int **SOMEBITS**

This flag in the infoflags argument to imageUpdate indicates that more pixels needed for drawing a scaled variation of the image are available. The bounding box of the new pixels can be taken from the x, y, width, and height arguments to the imageUpdate callback method.

**See Also:**[Graphics.drawImage(java.awt.Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.SOMEBITS)

### FRAMEBITS

static final int **FRAMEBITS**

This flag in the infoflags argument to imageUpdate indicates that another complete frame of a multi-frame image which was previously drawn is now available to be drawn again. The x, y, width, and height arguments to the imageUpdate callback method should be ignored.

**See Also:**[Graphics.drawImage(java.awt.Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.FRAMEBITS)

### ALLBITS

static final int **ALLBITS**

This flag in the infoflags argument to imageUpdate indicates that a static image which was previously drawn is now complete and can be drawn again in its final form. The x, y, width, and height arguments to the imageUpdate callback method should be ignored.

**See Also:**[Graphics.drawImage(java.awt.Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.ALLBITS)

### ERROR

static final int **ERROR**

This flag in the infoflags argument to imageUpdate indicates that an image which was being tracked asynchronously has encountered an error. No further information will become available and drawing the image will fail. As a convenience, the ABORT flag will be indicated at the same time to indicate that the image production was aborted.

**See Also:**[imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.ERROR)

### ABORT

static final int **ABORT**

This flag in the infoflags argument to imageUpdate indicates that an image which was being tracked asynchronously was aborted before production was complete. No more information will become available without further action to trigger another image production sequence. If the ERROR flag was not also set in this image update, then accessing any of the data in the image will restart the production again, probably from the beginning.

**See Also:**[imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.ImageObserver.ABORT)

| **Method Detail** |
| --- |

### imageUpdate

boolean **imageUpdate**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int infoflags,  
 int x,  
 int y,  
 int width,  
 int height)

This method is called when information about an image which was previously requested using an asynchronous interface becomes available. Asynchronous interfaces are method calls such as getWidth(ImageObserver) and drawImage(img, x, y, ImageObserver) which take an ImageObserver object as an argument. Those methods register the caller as interested either in information about the overall image itself (in the case of getWidth(ImageObserver)) or about an output version of an image (in the case of the drawImage(img, x, y, [w, h,] ImageObserver) call).

This method should return true if further updates are needed or false if the required information has been acquired. The image which was being tracked is passed in using the img argument. Various constants are combined to form the infoflags argument which indicates what information about the image is now available. The interpretation of the x, y, width, and height arguments depends on the contents of the infoflags argument.

The infoflags argument should be the bitwise inclusive **OR** of the following flags: WIDTH, HEIGHT, PROPERTIES, SOMEBITS, FRAMEBITS, ALLBITS, ERROR, ABORT.

**Parameters:**img - the image being observed.infoflags - the bitwise inclusive OR of the following flags: WIDTH, HEIGHT, PROPERTIES, SOMEBITS, FRAMEBITS, ALLBITS, ERROR, ABORT.x - the *x* coordinate.y - the *y* coordinate.width - the width.height - the height. **Returns:**false if the infoflags indicate that the image is completely loaded; true otherwise.**See Also:**[WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [Image.getWidth(java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Image.html#getWidth(java.awt.image.ImageObserver)), [Image.getHeight(java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Image.html#getHeight(java.awt.image.ImageObserver)), [Graphics.drawImage(java.awt.Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ImageObserver.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/image/ImageFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/ImageProducer.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/ImageObserver.html)    [**NO FRAMES**](http://docs.google.com/ImageObserver.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#35nkun2) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).