Interface java.awt.image.ImageProducer

public interface **ImageProducer**
extends Object

The interface for objects which can produce the image data for Images. Each image contains an ImageProducer which is used to reconstruct the image whenever it is needed, for example, when a new size of the Image is scaled, or when the width or height of the Image is being requested.

See Also:
ImageConsumer
Version:
1.8 09/08/95
Author:
Jim Graham

Method Index

o **addConsumer**(ImageConsumer)
This method is used to register an ImageConsumer with the ImageProducer for access to the image data during a later reconstruction of the Image.
o **isConsumer**(ImageConsumer)
This method determines if a given ImageConsumer object is currently registered with this ImageProducer as one of its consumers.
o **removeConsumer**(ImageConsumer)
This method removes the given ImageConsumer object from the list of consumers currently registered to receive image data.
o **requestTopDownLeftRightResend**(ImageConsumer)
This method is used by an ImageConsumer to request that the ImageProducer attempt to resend the image data one more time in TOPDOWNLEFTRIGHT order so that higher quality conversion algorithms which depend on receiving pixels in order can be used to produce a better output version of the image.
o **startProduction**(ImageConsumer)
This method both registers the given ImageConsumer object as a consumer and starts an immediate reconstruction of the image data which will then be delivered to this consumer and any other consumer which may have already been registered with the producer.

Methods

o **addConsumer**

```
public abstract void addConsumer(ImageConsumer ic)
```

This method is used to register an ImageConsumer with the ImageProducer for access to the image data during a later reconstruction of the Image. The ImageProducer may, at its discretion, start delivering the image data to the consumer using the ImageConsumer interface immediately, or when the next available image reconstruction is triggered by a call to the startProduction method.
See Also:
startProduction

o **isConsumer**

```
public abstract boolean isConsumer(ImageConsumer ic)
```

This method determines if a given ImageConsumer object is currently registered with this ImageProducer as one of its consumers.

o **removeConsumer**

```
public abstract void removeConsumer(ImageConsumer ic)
```

This method removes the given ImageConsumer object from the list of consumers currently registered to receive image data. It is not considered an error to remove a consumer that is not currently registered. The ImageProducer should stop sending data to this consumer as soon as is feasible.

o **startProduction**

```
public abstract void startProduction(ImageConsumer ic)
```

This method both registers the given ImageConsumer object as a consumer and starts an immediate reconstruction of the image data which will then be delivered to this consumer and any other consumer which may have already been registered with the producer. This method differs from the addConsumer method in that a reproduction of the image data should be triggered as soon as possible.
See Also:
addConsumer

o **requestTopDownLeftRightResend**

```
public abstract void requestTopDownLeftRightResend(ImageConsumer ic)
```

This method is used by an ImageConsumer to request that the ImageProducer attempt to resend the image data one more time in TOPDOWNLEFTRIGHT order so that higher quality conversion algorithms which depend on receiving pixels in order can be used to produce a

better output version of the image. The ImageProducer is free to ignore this call if it cannot resend the data in that order. If the data can be resent, then the ImageProducer should respond by executing the following minimum set of ImageConsumer method calls:

```
ic.setHints(TOPDOWNLEFTRIGHT | <otherhints>);
ic.setPixels(...);      // As many times as needed
ic.imageComplete();
```

See Also:

setHints