Core Image / CIImageProcessorKernel / formatForInput(at:)

Type Method

formatForInput(at:)

Override this class method if you want your any of the inputs to be in a specific pixel format.

iOS 10.0+ | iPadOS 10.0+ | Mac Catalyst 13.1+ | macOS 10.12+ | tvOS 10.0+ | visionOS 1.0+

```
class func formatForInput(at inputIndex: Int32) -> CIFormat
```

Discussion

The format must be one of `kCIFormatBGRA8`, `kCIFormatRGBAh`, `kCIFormatRGBAf` or `kCIFormatR8`. On iOS 12 and macOS 10.14, the formats `kCIFormatRh` and `kCIFormatRf` are also supported.

If the requested inputFormat is `0`, then the input will be a supported format that best matches the rendering context's `workingFormat`.

If a processor wants data in a colorspace other than the context's working color space, then call `matchedFromWorkingSpace(to:)` on the processor input. If a processor wants it input as alpha-unpremultiplied RGBA data, then call `unpremultiplyingAlpha()` on the processor input.

See Also

Type Methods

```
class func apply(withExtent: CGRect, inputs: [CIImage]?, arguments: [
String : Any]?) throws -> CIImage
```

Call this method on your Core Image Processor Kernel subclass to create a new image of the specified extent.

```
class func process(with: [any CIImageProcessorInput]?, arguments: [
String : Any]?, output: any CIImageProcessorOutput) throws
```

Override this class method to implement your Core Image Processor Kernel subclass.

```
class func roi(forInput: Int32, arguments: [String : Any]?, outputRect:
CGRect) -> CGRect
```

Override this class method to implement your processor’s ROI callback.

```
class func roiTileArray(forInput: Int32, arguments: [String : Any]?,
outputRect: CGRect) -> [CIVector]
```

Override this class method to implement your processor’s tiled ROI callback.