

ImageReaderService

Image Reader Service

DISCLAIMER: This service is not yet part of a Clerezza release. See also [CLEREZZA-716](#).

Purpose

By default the Clerezza Platform uses `javax.imageio` to read image data. However there are at least two scenarios when users may want to change that.

1. When reading images that `ImageIO` does not support (different image formats or image formats that may be considered corrupt by `ImageIO`)
2. When wanting to improve image processing performance.

A major difficulty in resolving these problems is that normally there is no way other than changing source code and recompiling to make existing services use a different method of reading images. With the introduction of the [ImageReaderService](#) all Clerezza Platform components have been changed to use the [ImageReaderService](#) when obtaining [BufferedImage](#)s. This service dynamically looks up services that implement the [ImageReader](#) interface in the OSGi SCR (Service Component Runtime) and provides the service that is registered with the highest `service.ranking` property. By default the Clerezza Platform has only one [ImageReader](#) service registered that has a service ranking of 0 and reads images using `javax.imageio`. The [ImageReaderService](#) implements the [ImageReader](#) interface as well but delegates all calls to the highest ranking registered [ImageReader](#).

How to use the [ImageReaderService](#)

You can bind the [ImageReaderService](#) in an OSGi component by referencing it:

```
@Reference
private ImageReaderService imageReaderService;
```

Then you can either call the `getImageReader()` method to obtain the highest ranking [ImageReader](#) registered or use one of the `getBufferedImage` methods that will delegate to the highest ranking [ImageReader](#).

How to register a custom [ImageReader](#)

To register a custom [ImageReader](#) with service ranking of 1 (default is 0) you need to create a bundle providing an [ImageReader](#) service as follows:

```
@Component
@Service(ImageReader.class)
@Property(name=Constants.SERVICE_RANKING, intValue=1)
public class BetterImageReader implements ImageReader {
    // ...
}
```

When the component `metaType` is declared `true` and the `service.ranking` property is declared non-private, the `service.ranking` can be dynamically configured at runtime in the Felix Web Console.

```
@Component(metaType=true)
@Service(ImageReader.class)
@Property(name=Constants.SERVICE_RANKING, intValue=1, propertyPrivate=false)
public class BetterImageReader implements ImageReader {
    // ...
}
```