User Interface - Feature #12170

Java replacements for the System.Drawing imaging subset used from 4GL

11/15/2026 03:45 PM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#1 - 11/15/2026 03:46 PM - Greg Shah

Problem

GUI applications written for OpenEdge on Windows can use the .NET System.Drawing classes directly from 4GL to load, resize, combine and save images, and to get the icon Windows associates with a file:

```
USING System.Drawing.* FROM ASSEMBLY.

DEFINE VARIABLE oSrc AS System.Drawing.Bitmap   NO-UNDO.
DEFINE VARIABLE oDst AS System.Drawing.Bitmap   NO-UNDO.
DEFINE VARIABLE oGfx AS System.Drawing.Graphics NO-UNDO.

oSrc = NEW System.Drawing.Bitmap(cImageFile).
oDst = NEW System.Drawing.Bitmap(100, 100).
oGfx = System.Drawing.Graphics:FromImage(oDst).
oGfx:DrawImage(oSrc, 0, 0, 100, 100).
oDst:Save(cThumbFile, System.Drawing.Imaging.ImageFormat:Jpeg).
oGfx:Dispose().
oSrc:Dispose().
oDst:Dispose().

oIcon = System.Drawing.Icon:ExtractAssociatedIcon(cDocument).
oIcon:ToBitmap():Save(cIconFile, System.Drawing.Imaging.ImageFormat:Jpeg).
```

FWD's front end recognises .NET class references, and the analytics reports list them, but the conversion produces no code for them and there is no runtime support. There is no .NET runtime in a converted system, and none is needed for this subset: Java 2D (BufferedImage, Graphics2D, ImageIO) does all of it, headless, on the server.

Work

1. Provide FWD replacement classes for these .NET members, with the same names, signatures and behaviour:
   - System.Drawing.Bitmap: Bitmap(character) (load from file), Bitmap(integer, integer) (new blank image), Save(character), Save(character, ImageFormat), Width, Height, Dispose();
   - System.Drawing.Image: FromFile(character), Width, Height, Save(character, ImageFormat);
   - System.Drawing.Graphics: FromImage(Image), DrawImage(Image, integer, integer), DrawImage(Image, integer, integer, integer, integer) (scaled), DrawImage(Image, Point), Dispose();
   - System.Drawing.Imaging.ImageFormat: Jpeg, Gif (and Png and Bmp, which cost nothing extra);
   - System.Drawing.Point: Point(integer, integer);
   - System.Drawing.Icon: ExtractAssociatedIcon(character), ToBitmap().
2. Extend the conversion so that USING System.Drawing.* FROM ASSEMBLY and references to these types resolve to the replacement classes, in the same way as Direct Java Access. A reference to a System.Drawing member not in the list above converts with a warning and fails at run

time with a clear message, rather than being silently dropped.
3. Dispose() and DELETE OBJECT release the Java image; they must not fail if called twice.
4. Icon:ExtractAssociatedIcon: there is no Windows shell icon store on the server. Ship a set of icons keyed by file extension (PDF, the Office formats, images, text, HTML, archives, and a generic default), resolve the icon from the file name's extension, and allow a project to add or override icons in its configuration. The file must exist, as in .NET: a missing file raises the same error.
5. File paths go through the same path mapping as other 4GL file access, so Windows-style paths in the code resolve on the server.
6. Image formats: JPEG output uses a quality setting close to .NET's default, and GIF output is palette-based. Note in the documentation that the output bytes will not match .NET exactly; size and visual content will.
7. Errors: a missing or unreadable file in Bitmap(character) or Image:FromFile raises an error the 4GL can trap with CATCH or NO-ERROR, matching the error OpenEdge raises when .NET throws.
8. Update the gap marking for these classes and members.

Testcases

- Load a JPEG, GIF, PNG and BMP; read Width and Height; save in each output format; compare dimensions and pixel content (within a tolerance) with the output of the same 4GL program on Windows OpenEdge.
- Create a blank bitmap, draw a scaled image and an unscaled overlay at a point, and save. Compare with Windows OpenEdge.
- Combine several images side by side into one new bitmap sized from their widths and the largest height.
- ExtractAssociatedIcon for each shipped extension, an unknown extension, an extension overridden in the configuration, and a missing file.
- Dispose() twice, DELETE OBJECT after Dispose(), and a missing input file trapped with CATCH and with NO-ERROR.