Managing High-DPI Behavior in isCOBOL UIs (Windows and Multi-Platform Options)

Starting with Java 11, the Java runtime is DPI-aware by default.
On displays configured above 96 DPI (for example 120, 150 or 200 DPI), Java automatically scales UI elements.

For standard isCOBOL GUI applications this may result in:

- resized UI layouts
- bitmaps or icons that become distorted or “garbled”
- differences between development machines (often 96 DPI) and end-user environments (120+ DPI)

isCOBOL offers two supported approaches to control DPI behavior, depending on platform requirements.

Option 1: Use non-DPI-aware Windows launchers (Windows only)

Beginning with isCOBOL 2024 R1, additional executable wrappers are available under the sub-folder binno_dpi_aware located in the SDK installation folder:

```
binno_dpi_aware
  iscclient.exe
  iscrun.exe
  isclient.exe
  isrun.exe
```

These wrappers:

- remove the dpiAware attribute from the executable manifest
- force sun.java2d.dpiaware=false internally
- prevent Java from performing high-DPI scaling

Effect on the UI

- icons appear consistent (no DPI distortion)
- the entire window appears slightly blurred, because Windows applies bitmap-based scaling

Limitations

- Supported on Windows only
- Window content scaling applies to child processes started through C$SYSTEM, C$DESKTOP, etc.
- The wrappers must be executed from their own directory; do not copy them over the original bin executables.
  Instead, either:
  - add binno_dpi_aware to the beginning of the PATH environment variable
  - reference the executable wrappers explicitly in the command line

Option 2: Disable Java UI scaling (multi-platform)

A platform-independent alternative is to add the following to the Java command line:

-Dsun.java2d.uiScale.enabled=false

for example via:

- startup scripts
- launcher configuration
- JVM options

Effect on the UI:

- UI elements do not scale at all
- the GUI appears exactly as it would on a 96-DPI display
- icons remain clean, never distorted

Limitation:

- UI may appear too small for users on high-DPI displays (e.g., 150 DPI laptop panels)

Supported platforms

- Windows
- Linux
- macOS

Comparison Summary

Method	Java Support	Platforms	Result

Windows no-DPI executables	Since Java 8	Windows only	Slight blur; stable layout; no icon distortion
-Dsun.java2d.uiScale.enabled=false	Since Java 11	Win / Linux / Mac	Sharp UI; no scaling; windows remain very small on high DPI

Recommendations

Use the method that matches your deployment needs and/or constraints:

- Windows-only shops wanting consistent UI >> use the no-DPI-aware wrappers
- Mixed OS environments >> apply the Java UI disable flag
- Users with 4K laptops or 150-200 DPI displays may prefer the Windows blurred option over very small windows

If you distribute multiple launch modes, we suggest documenting clearly which launcher is intended for high-DPI systems.

Online URL: https://support.veryant.com/phpkb/article.php?id=357