How to deploy bitmaps when running in the Application Server/Thin Client

When developing a graphical screen with bitmaps– either using the screen painter in the isCOBOL IDE are coding by hand – it is common to add bitmaps with a relative path. The bitmaps are found if the program is running in stand-alone mode, but aren't found when running in thin client.

In order to make your GUI application's bitmaps available in any environment, you can do one of these options:

1. Use a relative path and start iscserver from the same folder as the bitmaps
2. Put the full path to the bitmaps in the CLASSPATH (for instance, in isserver.vmoptions when running a windows service)
3. Use COPY RESOURCE to put a copy of the bitmap in the .class file
4. Use a configuration variable for the root directory, then pick up the variable and assemble the full path to pass to W$BITMAP

Options #1 and #2 are useful for development, but not practical for deployment. Options #3 and #4 are the suggested best practice options.

Option #3 was introduced in isCOBOL 2022R1. It embeds the bitmap in the class file, and can be accessed with wbitmap-load, . Here's an example of that code:

```
Copy resource "..resourceslogo.png".
call "w$bitmap" using wbitmap-load, "logo.png"
            giving h-bmp.
```

Option #4 requires access to the bitmap files external to the classes. This option gives you the most flexibility by using a configuration variable for the root path. The name of the variable is your choice, but must be preficed by 'iscobol'. For instance:

```
iscobol.bitmap1_path=c:myappbitmaps
```

Your code can then access this variable and build a path for the location of the bitmap:

```
accept bitpath from environment bitmap1_path
string bitpath,
        “/bitmaps/mybitmap.bmp”,
      into full-bmp-path
call "w$bitmap" using wbitmap-load, full-bmp-path
giving h-bmp.
```

You can also load a bitmap from the client machine in thin client. This approach that has two considerations: It can be faster because the request doesn't need to communicate with the server, but the bitmaps need to be available on all the client machines. If you have a very large bitmap, this method might be preferable, as the bitmap will load faster.

```
call "W$BITMAP" using WBITMAP-LOAD-FROM-CLIENT
                      "my-logo.png"
                giving h-logo
```

Online URL: https://support.veryant.com/phpkb/article.php?id=314