



You Captured My Screen!



How often have you thought, “If BBJ just had a function to capture an image of my screen...”?

If you are like many developers, you have thought this more than once over the years. And, after all, how hard could it be to just grab the screen contents and stuff it in a buffer, right? Unfortunately, it is not that easy, but with BBJ® 14.0 and higher, you can easily program a number of once-difficult tasks in only a few lines of code.

Through advancements in the BBJ language, you can now capture screens and all BBJ windows to BBJImage objects in both the graphical (GUI) and browser (BUI) user interfaces. BASIS has added a new [ScreenCapture](#) class to its arsenal of helpful utility classes in the file `<BBJ Install>/utils/screencapture.bbj`. The ScreenCapture class greatly simplifies your use of this UI-independent capturing functionality, giving you programmatic control over how to capture images.

In addition, we fine-tuned the existing [BBWindowUtils::centerWindow\(\)](#) method to take advantage of new [BBJTopLevelWindow](#) “getOuter” methods that deliver the outermost X and Y coordinates and dimensions of a window. The new versions of these methods take into account any window decorations such as a menu bar, window title, and frame. Until now, BBJ developers had to come up with their own algorithms to compute these values the best they could.

Enhancements to the BBJ Language

At the heart of the ScreenCapture class lie two capture-related methods that each return a [BBJImage](#) object.

1. [BBJSystemMetrics::getScreenImage\(\)](#) captures an image of the entire screen.
2. [BBJWindow::getWindowImage\(\)](#) captures an image of a top-level or child window.

Once you have a BBJImage object, you can manipulate it with code or stream it to a file.



By Ralph Lance
Software Developer

The `getOuter` methods added to the `BBjTopLevelWindow` class include the following:

- `BBjTopLevelWindow::getOuterX()` retrieves the outer window's X coordinate
- `BBjTopLevelWindow::getOuterY()` retrieves the outer window's Y coordinate
- `BBjTopLevelWindow::getOuterWidth()` retrieves the outer window's width dimension
- `BBjTopLevelWindow::getOuterHeight()` retrieves the outer window's height dimension

These methods allow you to position your windows more accurately, calculate the sizes and resize them, or perform a host of other previously complex actions where you need to know exactly what is where on the screen.

Screen Capturing Made Easy!

The `ScreenCapture` class installs with BBJ 14.0 and higher as a utility in `<BBj Install>/utils/screencapture.bbj`. It contains several static methods that you can invoke directly without having to first create (aka instantiate) an object instance of the class. `ScreenCapture` uses neither a client object instance of the popular `java.awt.Toolkit` for screen captures nor `java.awt.Robot` for window captures, so you can use it in both BUI and GUI programs.

Simply running `screencapture.bbj` brings up a demonstration window as seen in **Figure 1** that allows you to perform a screen, top level window, or child window capture.

By default, the demo program stores each capture image in a file in the 'Default temp directory' displayed near the top of the window. Marking the 'Show Save Dialog' checkbox gives you a 'File Save' dialog so you can override this default directory and store each image wherever you like. After saving your GUI capture to a file, it appears in the default image viewer on the client via a call to `BBjThinClient::browse()`. Captures made in BUI work identically, but their image file is made available to the web server via `BBUtils::copyFileToWebServer()`.

The demo code is at the bottom of the `screencapture.bbj` file, immediately following the classend statement for the `ScreenCapture` Class. The static methods in `ScreenCapture` allow you to use default values for pretty much everything. For example, you can use `ScreenCapture.capture([window!])` to use all of the default values, or you can set the image file path and name explicitly and specify whether the save file dialog should be presented using code like this:

```
ScreenCapture.capture([window!], temp_dir$ + "mycapture.png", 1)
```

The image file format used by `BBjImage::getBytes()` usually PNG, JPG, or GIF, is implied by the extension of the file name you provide.

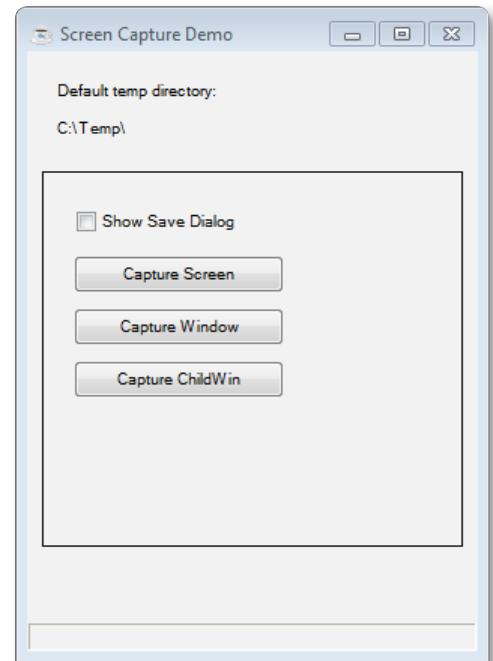


Figure 1. Screen Capture Demo

Summary

Due to the program control we now have, you can easily build screen and window captures into BBJ applications. One example would be in a common error handler routine that captures the state of the screen at the time of an error. Another use for this handy tool could be to "take a picture" of a top-level form or any part of a form contained on a child window programmatically while automatically creating rudimentary documentation and/or help files. The new `BBjTopLevelWindow` "getOuter" methods allow you to get the outermost coordinates and dimensions of a window without concerning yourself with window decorations.

These are just a couple of the most recent advances in the BBJ language that help BASIS create and enhance useful utilities for the BBJ community while giving you, the BBJ developer, more control with less effort. The possibilities for these new tools are limited only by your imagination! ■



Missed an Issue?

Advantage

www.basis.com/advantage