

Let's See Your

"Hello, Mr. Hancock. I have two Chile Lovers t-shirts and one Salsa Gift pack for you today. Would you please just sign here?"

Modern applications in today's paperless world often require a way to capture signatures on touchscreen-enabled devices. Whether for proof of delivery, service work orders, sales forms, or contracts, capturing handwritten text has become a necessity for today's businesses. **Figure 1** shows the Signature Capture demo running on an Android tablet and **Figure 2** shows the signature panel.

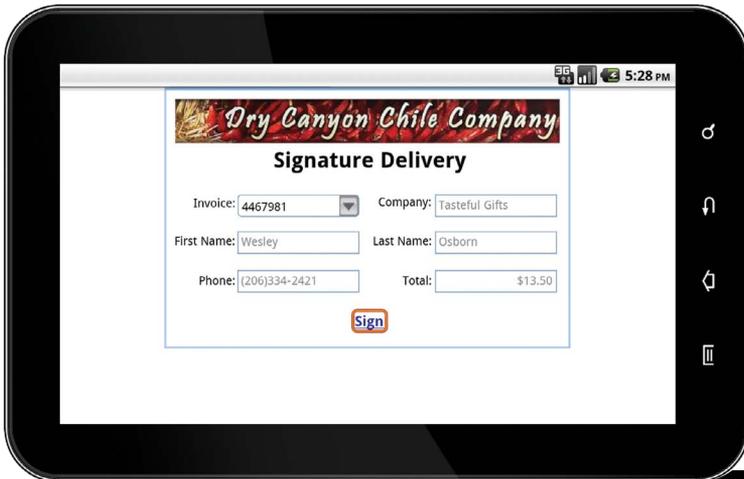


Figure 1. Signature Capture demo on an Android tablet



Figure 2. Draw panel for signature capture

John Hancock was president of Congress when the Declaration of Independence was adopted and signed. He is primarily remembered by Americans for his large, flamboyant signature on the Declaration, so much so that "John Hancock" became, in the United States, an informal synonym for *signature*.

The `setScribble()` method, new in BBj® 11.0, has an easy-to-implement mechanism of creating and storing handwritten text. Various supported platforms include laptops, smartphones, and tablets running on a variety of operating systems including Windows, iOS, Android, and Linux.

Available for all three BBj client flavors – GUI, BUI desktop, and BUI mobile/touch – getting and storing your John Hancock has never been so simple or secure.

The three primary methods to create and store a signature are `setScribble()`, `getDrawPanelImage()`, and `getBytes()`.

- `setScribble()` is used on the window (draw panel) to intercept relevant mouse or touch events and draws using the current PENWIDTH, PENCOLOR, PATTERN, and DRAWMODE settings.

```
#SignatureWindow!.setScribble(1)
```

- `getDrawPanelImage()` is used on the window or child window object. This method returns the draw panel as a `BBjImage` object.

```
sigImage!=#SignatureWindow!.getDrawPanelImage()
```



By Robert Del Prete
Quality Assurance
Engineer

- `getBytes()` is used on the `BBjImage` object to return raw image bytes that can then be written to disk as an image file. Image formats supported are `.png`, `.gif`, and `.jpg`. This file can be saved to the client, server, or stored in a database using a BLOB (Binary Large Object) datatype.

```
sigImage$ = sigImage!.getBytes("png")
```

The code snippet in **Figure 3** demonstrates getting the image from the draw panel. The first section gets information from a list box to name the resultant `.png` file. We then create a new `java.io.File` for the image. The `getDrawPanelImage()` method is used to get the draw panel as a `BBjImage` object and `getBytes()` allows the image to be written to disk.

```
rem Callback routine that is called when Save is pressed
method public void OnSavePressed(BBjButtonPushEvent p_event!)
  declare BBjNumber lbIndex!
  declare BBjString lbValue!
  declare BBjString sigFile$
  declare java.io.File sigFile!

  lbIndex! = #invoiceLB!.getSelectedIndex()
  lbValue! = #invoiceLB!.getItemAt(lbIndex!)

  sigFile$ = dsk("") + dir("") + lbValue! + "-signature.png"
  sigFile! = new java.io.File(sigFile$)
  sigImage! = #SignatureWindow!.getDrawPanelImage()
  sigImage$ = sigImage!.getBytes("png")

  image=unt
  open(image,mode="O_CREATE,O_TRUNC")sigFile$
  writerecord(image)sigImage$
  close(image)
  #SignatureWindow!.setScribble(0)
  #SignatureWindow!.setVisible(0)
  #this!.setDeliverButton()
methodend
```

Figure 3. Retrieving the image from the draw panel

Summary

While the need for the handwritten word has greatly changed since the time of John Hancock, there is still great need for capturing handwritten signatures electronically. Signature capture is available anywhere `BBj` runs, even on popular iOS and Android smart phones. All that is needed is the browser of any touch-enabled iOS or Android device to capture signatures in the field and to store the resultant image file safely on the server. Using `setScribble()` pays back big dividends in both time-savings and convenience. Digitally capturing signatures in the office, warehouse, or even on the go using mobile handheld devices is easy using the `setScribble()` method. ■

*"Thank you, Mr. Hancock.
Have a pleasant day!"*



For more information, refer to `setScribble()`, `getDrawPanelImage()` and `getBytes()` in the online documentation at links.basis.com/basisdocs

BARISTA®

... the ultimate RAD application generator for database-driven software projects and `BBj`®, the dynamic object-oriented programming language for Java

RABBIT APPLICATION DEVELOPMENT



Barista gets you (re)productive in no time.

Brought to you by

