

BBj Form Printing

By Jason Foutz



The new BBjPrinter system provides developers with a powerful new Application Programming Interface (API) for creating forms. This unified system makes it easy to add images to forms and simplifies the programmatic creation of documents containing letterheads and logos. The BBjPrinter object provides methods for laying out text, lines, and images on a page. In addition, developers can now effortlessly use different fonts on the same page with this new API.

The new printer system provides a complete **object hierarchy** that gives developers control over every aspect of printing - from **selecting printers** to how a line appears on a page. The **BBjPrinter object** allows the user to select a printer for BBj® Services to use and start printing. BBjPrinter allows developers to obtain a **BBjPrintDoc**, which represents the current document the program creates. The developer can then preview or print this document.

A **BBjPrintPage** represents a single page of the BBjPrintDoc. The BBjPrintPage object provides methods for creating and laying out lines, images, and blocks of formatted text. Using a BBjPrintPage, programmers can now create **BBjPrintLine**, **BBjPrintImage**, or **BBjPrintParagraph** objects.

The first step to using the new print system is to obtain a BBjPrinter. The BBjPrinter object represents a printer that BBj interacts with and provides configuration properties such as paper size, margin width, and orientation. Obtain the default printer object using the calls below:

```
LET myAPI!=BBjAPI()  
REM Obtain an instance of a default BBjPrinter object  
LET myBBjPrinter!=myAPI!.getBBjPrinter(0)
```

After obtaining a BBjPrinter, developers can then query for a list of all available printers on the servers or the client. Moreover, developers can change this instance of BBjPrinter to represent any other printer.

The second step is to create a BBjPrintDoc. Using the BBjPrinter object, obtain a new BBjPrintDoc using this call:

```
REM Get BBjPrintDoc for the selected printer  
LET myBBjPrintDoc!=myBBjPrinter!.getPrintDoc(1)
```

The newly created BBjPrintDoc represents an empty document with which to work. This document represents a collection of pages, all of which are available for preview or printing.

The third step is creating a new page to work with by using this call:

```
REM Get a new page  
LET myBBjPrintPage!=myBBjPrintDoc!.getNewPage()
```

This call creates the page; it does not add the page to the document. To add the page to the document, use this call:

```
REM Add the page to the document  
myBBjPrintDoc!.addPage(myBBjPrintPage!)
```

Pages are the foundation of forms. Developers can add the page at any time and they print in the order of addition. Furthermore, the pages created can contain BBjPrintObjects such as BBjPrintParagraph and BBjPrintLine.

The BBjPrintObject provides all of the methods for laying out a page, such as **setPosition** and **setSize**. In addition to the other methods available in the BBjPrintObject, these two methods provide complete control over page layout. BBjPrintParagraph, BBjPrintLine, and BBjPrintImage all extend BBjPrintObject and their layout uses the same methods included in the BBjPrintObject.

[continued...](#)

After creating a page, the next step is to add a sampling of BBJPrintObjects to this page. The code sample below shows how to add text, construct a paragraph object, set its text, and add the paragraph back to the page.

```
REM Get a new paragraph
LET para!=myBBJPrintPage!.newParagraph()

REM Set paragraph text
para!.setText("some sample text")

REM Add print object to page
myBBJPrintPage!.add(para!)
```

The BBJPrintParagraph object automatically fits the text into the available space. The BBJPrintParagraph adjusts its height automatically to take up as much vertical space as necessary to fit all of the text on the page with a call to this method:

```
REM Set paragraph auto adjust height
para!.setAutoAdjustHeight(1)
```

The paragraph object provides control over how the text flows on a page. If automatic height causes the text to use more space than looks appropriate, developers can set the size explicitly. In addition to sizing, paragraph justification is also available. For example, use the call below to make the text left justified:

```
REM Set paragraph horizontal alignment
para!.setHorizontalAlignment(para!.LEFT_JUSTIFIED)
```

To add lines to the page, use this call:

```
REM Get a new line
LET line!=myBBJPrintPage!.newLine()
line!.setPosition(new java.awt.Point(72,72))
line!.setEndPoint(new java.awt.Point(144,144))
REM Add print object to page
myBBJPrintPage!.add(line!)
```

This call creates a simple diagonal black line. Use the API calls to set various properties such as thickness.

The final step looks at everything as a whole using the call:

```
REM Print the document
myBBJPrintDoc!.preview()
```

continued...

Alternatively, use the following call to print the document:

```
REM Print the document  
myBBjPrintDoc!.print()
```

This article described how to construct a BBjPrinter object and used it to gather information about the printer. A BBjPrintDoc was used to create and preview a BBjPrintPage. This article also explained how to use the BBjPrintPage to construct several different kinds of BBjPrintObjects. Using this system, BASIS created the Purchase Requisition shown in **Figure 1** (see **sample code**). The new BBjPrinter system gives developers a variety of powerful new tools to create professional forms. Developers can now use the uniform BBjPrintObject to add images, lines, and text to their forms.

The screenshot shows a 'Print Preview' window for a 'Purchase Requisition' form. The form is titled 'BASIS PURCHASE REQUISITION' and includes the following fields and sections:

- Requested By:** [Text Field] **Date:** [Text Field]
- Choose one:** Flat, Issue PO No PO required. Item(s) will be received in Shipping & Receiving. Off vendor requires a Purchase Order. This is approval backup for the purchase, no items will be received.
- Cost Center:** [Text Field] **Purpose:** [Text Field]
- Vendor:** [Text Field] **Contact:** [Text Field]
- PO #:** [Text Field] **Phone:** [Text Field]
- Table:** A large table with multiple rows and columns for item details.
- Payment:** Bill Us Basis Credit Card Check Enclosed
- Shipping:** 1 Day 2 Day 3rd N/A Pickup Deliv Electronic
- Arrival Date:** [Text Field]
- Comments:** [Text Field]
- Summary:** SubTotal \$, Freight \$, Tax \$, TOTAL \$
- Approvals:** Manager Approval _____ Finance Approval _____
- Footer:** Accounting: Above & Below Requisitioner: Pink

Figure 1. BASIS Purchase Requisition.