



New BBj Output Types, Including The Cloud!

In BBj® version 11.0, the BBJasper Utility is compatible with JasperReports created in iReport version 4 and supports additional industry standard save formats. In conjunction with the GApps Utility, developers and end users alike now have the ability to save generated reports to the cloud. Viewing a Jasper report via the BBJasperViewer is no longer a hands-off experience as the Utility now saves viewer defaults and can issue receive notifications to end users' save actions. Let's take a closer look into these enhancements.

Compatibility with iReport

New BBj releases include, as applicable, updated JasperReports jars to ensure compatibility with the latest version of iReport – the WYSIWYG JasperReports design tool. **Figure 1** shows release compatibility between BBj and iReport, both of which rely upon the underlying JasperReports library.

New Save Formats

In previous BBj versions, one could save a JasperReport in the following formats:

- jprint - Jasper Print Format
- pdf - Portable Document Format
- html - Hypertext Markup Language

Now, BBj 11.0 adds these formats to that list:

- xls - Excel Spreadsheet (single or multiple sheets)
- csv - Comma Separated Values
- rtf - Rich Text Format
- odt - Open Document Text (OpenOffice)
- xml - Extensible Markup Language >>

BBj 9.0	iReport 3.5.0
BBj 9.01	iReport 3.5.0
BBj 9.10	iReport 3.5.1
BBj 9.12	iReport 3.6.0
BBj 10.0	iReport 3.6.2
BBj 10.01	iReport 3.6.2
BBj 10.02	iReport 3.7.4
BBj 10.03	iReport 3.7.4
BBj 10.04	iReport 3.7.4
BBj 11.0	iReport 4.02
BBj 11.11	iReport 4.1.1

Figure 1. BBj and iReport version compatibility



By Brian Hipple
Quality Assurance
Supervisor

```

NetBeans IDE 3.6 - Project Default
File Edit View Project Build Debug Versioning Tools Window Help

rem Creates the sales by month report, exports it to spreadsheet on the client
rem Use statements
use ::bbjasper.bbjs::BBJasperReport

rem Declares
declare BBJAPI api!
declare BBJasperReport report!

rem Get a reference to the BBJAPI
api! = BBJAPI()

rem Get the BBJ Home directory
bbjHome$ = java.lang.System.getProperty("basis.BBJHome")

rem Set the report file
reportFile$ = bbjHome$ + "/demos/Reports/SALES_BY_MONTH.jasper"

rem Set the connect string
connectString$ = "jdbc:basis:localhost?database=ChileCompany&user=admin&password=admin123"

rem Create the report
report! = new BBJasperReport(reportFile$,connectString$)

rem Fill the report with data
report!.fill()

rem Export the report to an XLS file
report!.exportToXLS(api!.TRUE, "/tmp/test.xls")

release

6:24 INS
Test.bbjs saved.
    
```



Figure 2. Sample code that exports a report to Excel format

INVOICE	COMPANY	INVOICE DATE	ORIGINAL AMOUNT	DISCOUNT AMOUNT	ACTIVITY TO DATE
4467981	Tasteful Gifts	11/30/1993	\$ 13.50	\$ 0.00	\$ 0.00
4567801	FishLand	10/16/1993	\$ 240.08	\$ 0.00	\$ 0.00
6408901	Frame, Mary	12/20/1993	\$ 179.40	\$ 0.00	\$ 0.00
6458701	Aurora, Justin	09/14/1993	\$ 17.00	\$ 0.00	\$ 0.00
7123465	Gourmet Giftware	12/14/1993	\$ 215.00	\$ 0.00	\$ 0.00
7403456	El Pollo Caliente	10/14/1993	\$ 15.25	\$ 0.00	\$ 0.00
8790510	Condiments & Thensome	06/20/1993	\$ 116.59	\$ 0.00	\$ 0.00
8974445	Abbott, Meghan	12/13/1993	\$ 169.00	\$ 0.00	\$ 0.00
9704150	Riggle, Robert	12/14/1993	\$ 162.69	\$ 0.00	\$ 0.00
9902541	Massage Land	12/21/1993	\$ 86.00	\$ 0.00	\$ 0.00
Total Amount Due:				\$ 1,214.51	
Less Discount				\$ 0.00	
Less Payment				\$ 0.00	

Figure 3. BBJasperViewer new Save options

An exportTo<format> method, shown in **Figure 2**, is available in the BBJasper Utility for each type listed above.

Saving to the Cloud

The GApps Utility provides an easy way to upload documents to the cloud. Calling the `GDocsService` method `uploadDocument` uploads a generated JasperReport for authorized users to share and view. A new menu button in BBJasperViewer (**Figure 3**) allows the end user to save the displayed report in the cloud to Google docs. After the user provides authentication to the Google docs via a login dialog, a save dialog appears in which the user selects the folder, file name, and type of document to save.

Setting Viewer Defaults and Getting Events

Previously at run time, the BBJ program created the report, filled it in with data, and subsequently handed it off to the BBJasperViewer for the user to view. The developer had no control over the default folder, file name, and file type that the save dialog presented to the

user. There was also no way for the BBJ application to know when the user saved the report, with BBJ 11.0 all of that was changed.

BBJ 11.0 introduces the following methods to set the default folder, file name, and file type for the both standard file saves and for saving to the cloud:

- `GSelectionDialog` (links.basis.com/pljmr) - `setDefaultFolder`, `setDefaultDocument`, `setDefaultType`
- `BBJasperViewerControl` (links.basis.com/dhcpt) - `setGoogleDefaultFolder`, `setGoogleDefaultDocument`, and `setGoogleDefaultType` >>

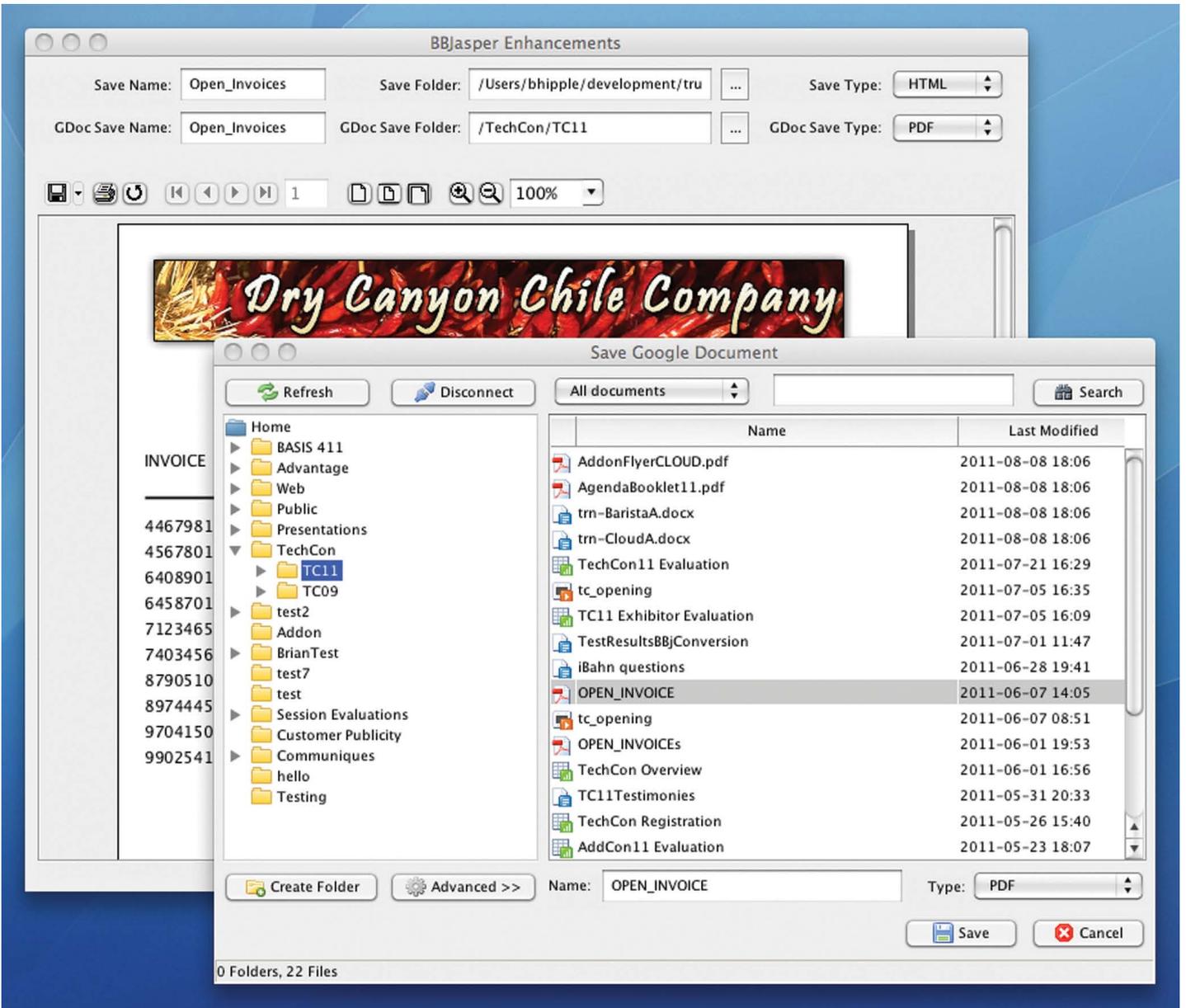


Figure 4. The Save dialog

The Save dialog shown in **Figure 4** allows the user to change these file specifications as desired.

Two additional methods, [setFileSaveCallback](#) and [setGoogleDocSaveCallback](#), register for events when the user saves the report to disk or to the cloud. These events contain the folder, file name, and file type of the saved report, a very useful capability for archiving purposes that Barista's Docout subsystem leverages.

Summary

Reporting is a very important part of almost every computer system. BASIS provides the BBJasper utility to facilitate easy integration of your BBJ application code with the iReport open source Java-based report design tool for the JasperReports standard. BASIS now extends the functionality of BBJasper to include five additional save formats. And, in keeping with the widespread adoption of the cloud, the BBJasper utility now also sports integration to the Google Cloud offerings. Another powerful building block utility for your world-class applications. ■



Read [Easy Integration To Google Apps](#)