DocOut Easily Modernizes BBx Reports

Add Print Preview, PDF, XLS, CSV, XML, Google Docs, Fax, Email, and Archiving

DocOut is the document output subsystem component of the Barista® Application Framework RAD tool with a ton of built-in benefits and features. Developers can now leverage these benefits from new or existing PRO/5®, Visual PRO/5® or BBj® code, without having to use the entire framework, and with very little programming effort!

To take advantage of DocOut, developers would simply change all of their `'PRINT @'` statements to vector assignments and invoke the DocOut

object. With those changes in place, applications are upgraded to a modern reporting system chock full of new features including print preview, multiple output format options, user interactive column sizing, report archiving, and more! After converting an application to use DocOut, a pleasing print preview of the report appears similar to the one shown in **Figure 1.**

Figure 1. DocOut report in print preview

By Ralph Lance
Software Engineer

Follow along in this review of the DocOut benefits and see the enhanced sample application.

Feature Overview

The DocOut report provides the same columnar or tabular report as the legacy program, but the DocOut system supercharges it with several extra capabilities. DocOut offers a number of output file types including PDF, XML, CSV, and XLS, shown in **Figure 2,** and puts the user in control.

After selecting the [Save] menu button, the user is able to choose the name of the saved report as well as the output type from multiple file formats. DocOut even provides an interface to Google Docs so users can save their reports in the cloud for ubiquitous access from any Internet-connected machine or device. Raw output is also an output option for those who use a third party tool to manipulate that output. Sending the report to the printer is a standard option, of course, but the DocOut system also allows the user to fax or email the report directly from the preview window.

In addition to viewing options such as scrolling through the report, sizing the output, and zooming in and out, the user is also able to affect the layout of the report directly from the preview window, as shown in **Figure 3**. DocOut gives the user the ability to adjust the widths of individual columns or even hide them if they are not applicable for the report's recipient. To aid in alignment for numeric data, DocOut can optionally modify the report to use a fixed-width font. Talk about the putting the user in control!

DocOut is flexible enough to give not only the user but also the application extensive control over the report. By adding code to automate DocOut's output settings, the app is in complete control over the report – even to the point of eliminating the user interface. This is a must-have feature for batch reports, as the application can configure and emit all of its reports without requiring any user intervention.

DocOut also offers a one-step methodology allowing the user to select and process a particular output type. Barista's Document Inquiry System is a couple of mouse clicks away, providing the ability to select multiple output types and reprint documents. Additional options are set directly from the Document Output Selection window (**Figure 4**) such as sending
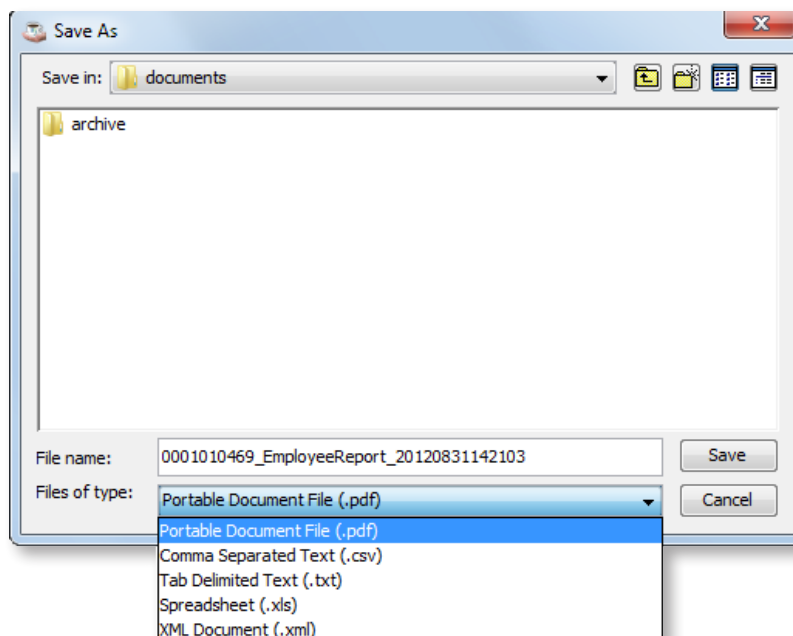

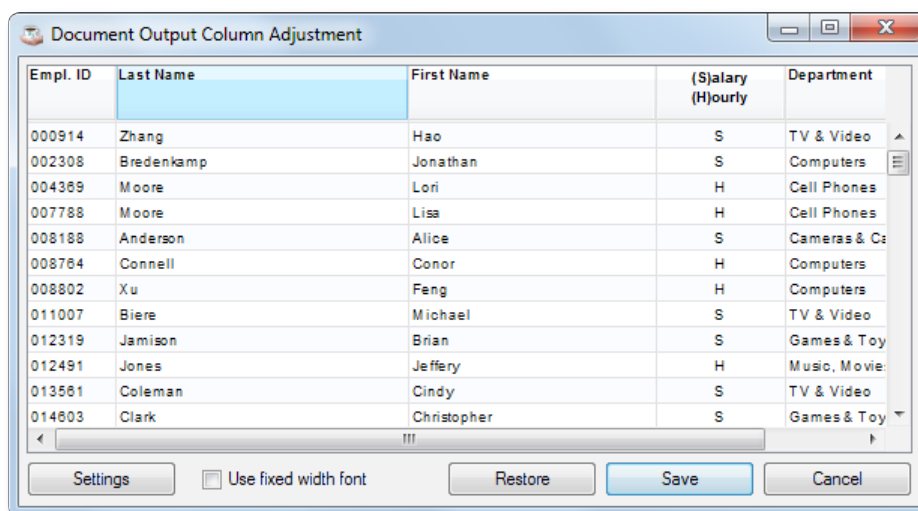**Figure 2**. Save As output formats

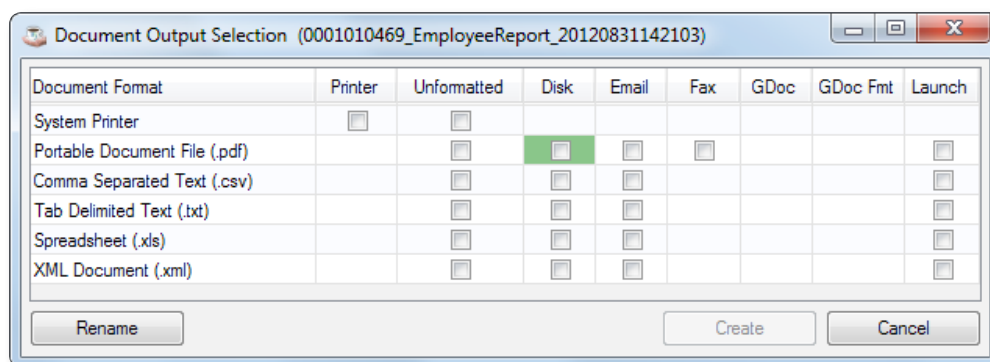**Figure 3.** DocOut's print preview with output column customization

Figure 4. Document Output Selection Dialog

the document via fax or email, saving the document in Google Docs (GDoc) in the cloud, and even launching the target document once the processing has completed.

Utilizing DocOut saves significant time and money in the development effort. Kurt Williams, of Marex Services acclaims:

> "It's very easy to convert an existing report to the DocOut process. Instead of using PRINT statements to send the data off to the printer, you just load the data into a BBj Vector and let DocOut process the report. The payoff is huge. The report is rendered in a print preview window and from there, you can adjust columns, output the report to the printer, convert it to a pdf, xls or delimited text file, email or fax it to interested parties, archive it, or simply review the data in the print preview and discard it!"

Using the DocOut Object

Everything needed to create amazing reports and output them in various types, including PDF and spreadsheet files, is built into the system without the need for adding plugins or prerequisites for third party products. Developers not only maintain complete customization control, but they benefit automatically from new features added to DocOut without requiring any additional code changes to their applications.

DocOut offers multiple benefits and incorporates an abundance of functionality. But just how much coding effort is required to take advantage all of this functionality? Let's take a look. BASIS designed the DocOut Object for use by any BBj program, obviating the need for the program to run inside the Barista framework. This approach allows legacy apps such as a CUI PRO/5 app running in BBj, for example, to have greater flexibility when it comes to reporting.

The new code is simple and very straightforward. For a more descriptive overview and detailed analysis of the process, review the DocOut Tutorial and the DocOut Object documentation referenced at the end of this article.

Our step-by-step example uses the DocOut object to produce a report of employees for a traditional character-based application called the CD-Jazz Store. Try the examples yourself by downloading the code from links.basis.com/12docout-code.

1. Modify the program to set up the Barista environment instead of opening the printer alias.

   a. Specify the complete path to the DocOut class when outside the Barista environment.

```
use ::C:/basis/barista/sys/prog/bao_document.bbj::Document
```

   b. Otherwise, use this relative path within the Barista environment.

```
use ::sys/prog/bao_document.bbj::Document
```

2. Declare object variables used in the program.

```
declare Document doc!
declare BBjVector out!
```

3. Create (instantiate) the DocOut Document object.

```
doc! = new Document()
```

4. Set the user authorization. Alternatively, use BBj User Authentication setup in Enterprise Manager.

```
doc!.setUserID("ADMIN")
doc!.setPassword("admin123")
```

5. Set the report parameters.

```
doc!.setFirmID("01")
doc!.setLanguage("ENU")
doc!.setDocumentID("EMPL_REPORT")
doc!.setReportTitle("Employee Report")
```

6. Set the report headings.

```
doc!.addReportHeading("Employee Report")
```

What's the next step? Set up the report columns.

The addColumn() method makes this a snap. The application code passes all of the necessary information for the report columns, including the following:

- Column heading
- Base data type
- Data length
- Optional column width override in pixels
- Optional output mask to be applied
- Optional Barista control type code (for future use)
- Optional justification code (defaults "L"eft for character, "R"ight for numeric columns)
- Optional formatting flags that can be used in combination, such as setting the typeface to bold and stipulating that the value is a total and should be underlined

Again, the DocOut tutorial and documentation describe these parameters and their permissible values in more detail. To localize the headings, take advantage of BBj's translation tools.

Based on the template for the employee record example, the code to set up the report columns is pretty straightforward:

```
doc!.addColumn("Empl. ID", "C", 6, 60, "", "", "", "")
doc!.addColumn("Last Name", "C", 50, 200, "", "", "", "")
doc!.addColumn("First Name", "C", 50, 200, "", "", "", "")
doc!.addColumn("(S)alary^(H)ourly", "C", 1, 60, "", "", "", "")
doc!.addColumn("Department", "C", 20, 200, "", "", "", "")
```

The use of the caret character in **"(S)alary^(H)ourly"** stipulates the desire for multiple column header lines.

With the report defined, it's time to fill it. In this example, simply read through the employees file and store the individual field values as strings in a BBjVector that gets passed to the **setOutputData()** method.

If the original application existed before DocOut, then it will most likely have contained **PRINT @** statements and page handling logic. As we can see here, DocOut integration greatly simplifies this code:

```
empl_chn = unt
open(empl_chn)"d:/training/cdjazz/data/cdj_empls"
dim empl$:"empl_id:c(6*),name_last:c(50*),name_first:c(50*),sal_hour:c(1*),dept:c(20*)"

out!=BBjAPI().makeVector()
while 1
   readrecord(empl_chn, end=*break)empl$
   out!.addItem(cvs(empl.empl_id$,3))
   out!.addItem(cvs(empl.name_last$,3))
   out!.addItem(cvs(empl.name_first$,3))
   out!.addItem(cvs(empl.sal_hour$,3))
   out!.addItem(cvs(empl.dept$,3))
wend
doc!.setOutputData(out!)
```

The final step is to instruct the DocOut object to actually produce the report and release our program with this code:

```
doc!.process()
release
```

By default, the DocOut object starts the report generation process synchronously so the program will wait for its completion. To override this, invoke the setSessionWait()method:

```
doc!.setSessionWait(0)
```

Using the DocOut Object With PRO/5 and Visual PRO/5

With the advent of XCALL, existing Visual PRO/5 and PRO/5 applications can also take advantage of the DocOut Object to XCALL the BBj program version of a legacy report. DocOut can either interact with the end user via a BBj thin client session on the desktop or can be invoked in BBj on the server-side only, with no user interface. In that case, one could still run legacy code in PRO/5 on a terminal emulator while generating the email/fax/pdf/archived document in BBj on the server. To view the report on a PC running PRO/5 or BBj in a terminal emulator, simply save it as a PDF in a directory on the server that the built-in Jetty Web Server can access, then provide the URL to the CUI session.

We encourage you to take a look at the documentation for the DocOut Object and the tutorial so you can retrofit your code by replacing those **PRINT @** statements with vector assignments and reap the multitude of DocOut benefits such as archiving, multiple file formats, and multiple delivery mechanisms, all with the minimum of programming effort. ■

- Download the code samples in this article at links.basis.com/12docout-code
- For more information about DocOut and the methods referenced in this article, see
  - DocOut Tutorial at links.basis.com/docout_tutorial
  - DocOut Object at links.basis.com/docoutobject
- Find BBjTranslator in the online documentation at links.basis.com/bbtranslator