

Desktop Data Delivered

By Adam Hawthorne



While earlier versions of BBJ[®] allowed access to the client's filesystem via a fat client deployment, access in a thin client deployment was not available. Frequently, developers expressed the desire for file access outside of fat client deployment; they wanted flexible and full access to files and directories anywhere in their client-server deployment.

BASIS offers two solutions. New MODE="CLIENT" options on the FILEOPEN and FILESAVE functions provide easy interaction with the client's filesystem from the traditional standard dialogs. However, when the application requires a customized dialog, or more flexibility choosing files and directories, enhancements in BBJ 7.0 expand the user's ability to interact with the client computer. The new BBJFileChooser control and BBJClientFileSystem object allow unprecedented interaction with the client's filesystem. This article focuses on the latter solution and the new freedom developers have to access information, anywhere, regardless of whether it is server- or client-based.

Introduction

The BBJClientFileSystem provides a straightforward interface to handle system-wide file actions. Using BBJClientFileSystem, the developer can obtain a specific file from the root filesystem, the user's home directory, or an absolute path to the desired file. Another new object, the BBJClientFile, provides the API for individual files. Its API models the Java class called `java.io.File`, which represents the name of a legal file on the client's filesystem. BBJClientFile objects may represent non-existent files or directories, and the API provides methods to create a non-existent file or directory. Most of the methods on the BBJClientFile match their respective methods on an instance of a `java.io.File`. These include self-explanatory methods such as `canRead`, `canWrite`, `isDirectory`, `exists`, etc. For more information about the other methods, refer to the BBJClientFile in the online documentation at www.basis.com.

Specifically, these four BBJClientFile methods allow data exchange between the client and the server:

RETURN VALUE	METHOD
string	BBJClientFile::copyFromClient()
void	BBJClientFile::copyToClient(string fileName\$)
string	BBJClientFile::getContents()
void	BBJClientFile::setContents(string contents\$)

To give these methods context, let us put them to work.

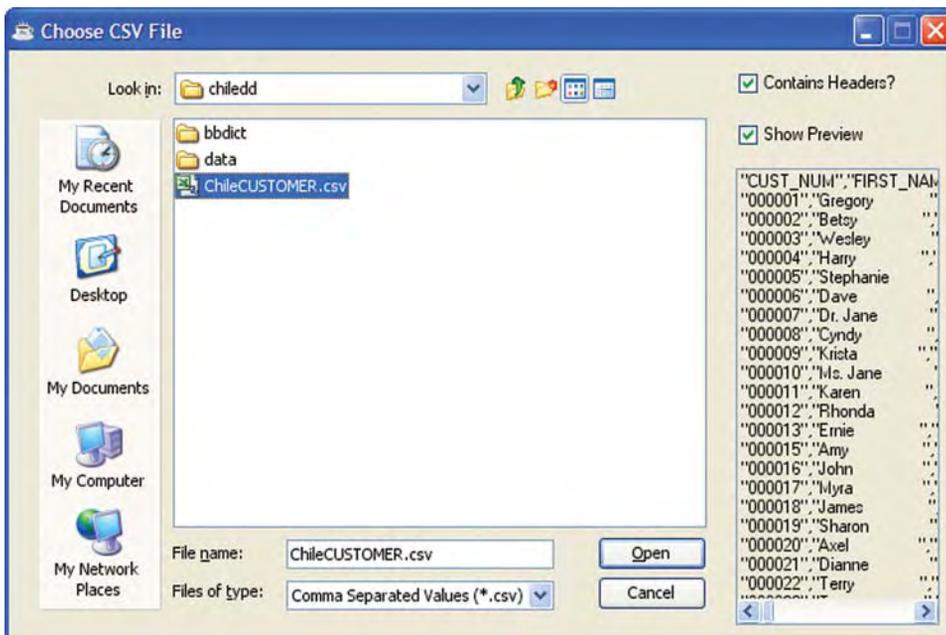


Figure 1. The customized file chooser dialog with file preview capabilities

Sample Program

To demonstrate the BBJClientFileSystem, look at the sample program `clientFSArticle.src`, downloadable from the URL noted at the end of this article. It allows users to select a CSV file on their own computer to display in a grid. It uses the aforementioned `getContents` and `copyFromClient` methods to obtain the contents of the file in two different contexts. When showing a customized client-side file chooser dialog (see **Figure 1**), the sample uses `getContents` to show a preview of the CSV file. It retrieves the data from the client and places it into the preview control on the file chooser dialog displayed on the client. This allows the user



Adam Hawthorne
Software Engineer

continued...

to see the contents of the file before selecting it to determine whether the file has a header row. This is a classic example of going above and beyond the standardized file open dialog; it extends the basic dialog with preview capabilities and other user options to provide all of the necessary information to display the CSV file.

Once the user selects a file, the application uses the `copyFromClient` method to copy the file from the client's machine to the server machine, thus obtaining the data in the file. When the sample calls `copyFromClient` on the `BBjClientFile` variable `file!`, `BBj` retrieves the data in the file named by the `file!` variable, copies it to a temporary file located on the server, and returns the name of the temporary file. When the copying operation completes, the program parses the file into a `CSVTable` object and displays the table in a grid on the client system, as shown in **Figure 2**. When the program terminates, it automatically deletes the temporary files that this method created. To retain a copy of the file on the server after the end of the program, use the `RENAME` verb to move the file to a different location.

CUST_NUM	FIRST_NAME	LAST_NAME	COMPANY	BILL_ADDR1	BILL_ADDR2	CITY
000001	Gregory	Baldrake	Bogus Stuff	8508 Manitoba NE		Albuq
000002	Betsy	Heebink	Betsy Inc	1499 378th		Madis
000003	Wesley	Osborn	Tasteful Gifts	3222 Gregory Street NE	P.O. Box 1997	Seattl
000004	Harry	Chuckie	Long Island Inc	9406 Sand Pebble Court		Palm :
000005	Stephanie	McIntyre		1712 Elm Terrace		Mode
000006	Dave	Strum	FishLand	6441 Concord Street		Hagei
000007	Dr. Jane	Booker	Greencastl Moose Farm	3898 Arcadia Trail East		Greer
000008	Cyndy	Sikes	Gourmet Giftware	421 Lawton Ave	Suite 2	DeFoi
000009	Krista	Nugent	Utopian Cattle	421 Navajo Trail NE		Albuq
000010	Ms. Jane	Ghast	Massage Land	1941 Wallace Avenue		Marcl
000011	Karen	Dennison	Rexall Drugs	3626 Ash Avenue NW	Apt 13	Albuq
000012	Rhonda	Frenchi	Bing Bong Gift co	5523 Russell Drive		Albuq
000013	Ernie	Augustine		3120 Carlos Court SE		Albuq
000015	Amy	Alcott	Misha Gifts	107 Cody Trail NW		Albuq
000016	John	Roberts	Bimland Ketchup	196 Ridge Road		Bellev
000017	Myra	Tomlin	Condiments & Thensome	397 Windsong Drive		Houst
000018	James	Thomas		P.O. Box 989		Albuq
000019	Sharon	Reeves		3107 84th Avenue		Seattl
000020	Axel	Pasternak		5405 Shelby Court		Albuq
000021	Dianne	Welleley		6802 Cassidy Lane		Santa
000022	Terry	Green		1631 Santa Maria		Middl
000023	Tom	Sanderson		6610 Spanish Elm Drive		El Pa
000024	Grover	Evanson		1120 Salem Avenue		Boise
000025	Doris	Williams		2900 Vista Del Rey		Puebl
000026	John	Tuscary	Monkey Shines	220 Rio Grande	Apt 16	Los Li
000027	Sally	Larimore		P.O. Box 2100		Portla
000028	Kathy	Nightengale		7121 College Ave		lthica
000029	Meghan	Abbott		8057 Tarheel		Raleig

Figure 2. The contents of the CSV file displayed in a grid

The majority of the code in this sample application comes from parsing the CSV file. The source code that shows the file chooser dialog and copies the file from the client, including the code to preview files and respond to events, is relatively small. The supplemental program, `csvLibrary.src`, provides an object-oriented model for a simple table and a class with methods to produce the table from a file.

Summary

BBj 7.0 gives developers who leverage these new API tools much greater flexibility in interacting with the client computer. They can store or retrieve configuration, data files, or even customized programs on or from the client computer using `BBjClientFileSystem` and `BBjClientFile`. Programs can deliver PDF files, spreadsheets or any other useful files directly to the user's computer at a location the user chooses using the `BBjFileChooser` in client mode and can retrieve configuration or other files as needed. Now, enjoy the freedom to place files where you need them using the `BBjClientFileSystem`. Deliver data, right to your desktop!



Download the sample code from
www.basis.com/advantage/mag-v11n1/clientfilesystem.zip