Scott Carter

4/22/2002

FlashFiler Basics

If you're not familiar with client/server database systems, it can be difficult to understand how to get started with FlashFiler. Once you're familiar with the general client/server database concepts, getting started is simple. This article describes those basic concepts and demonstrates how FlashFiler users perform some of the simple operations.

Client/Server Concepts

FlashFiler is a true client/server database system. This means that all requests to view tables, create tables, add records, query data, etc. are sent from a client application to a server application who actually carries out the task. The server receives the requests, performs the necessary actions, and then sends the results, if any, back to the client. Often, there are many clients sending requests to the server very rapidly. Since the server is the only application that actually manipulates data, it acts as the central manager for all table operations and ensures that the many requests are completed without compromising the integrity of the data.

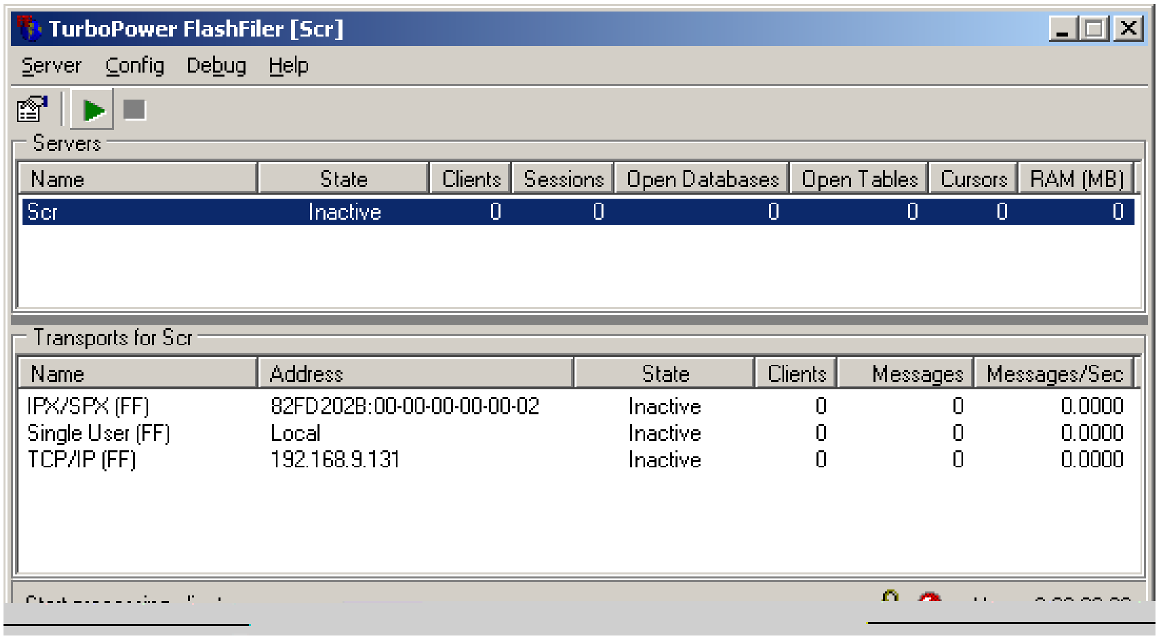
For comparison, database systems such as Paradox (BDE) and Microsoft Access are file-sharing systems. This type of database system accesses the tables directly. One problem with this type of system is that each client maintains its own copy of the data. When the clients update the data file there is a possibility that more than one client will try to update the same data at the same time. This can cause the table's data to be corrupted if some type of an anomaly, such as a power outage, occurs.

For developers converting from a file-sharing system to a client/server database system, the main thing to remember is that *client applications must have a database server application available to handle client requests*. The next section discusses server applications and the available options for FlashFiler users.

FlashFiler Server Applications

FlashFiler developers have several server options available to them. The most popular and simplest to use is FlashFiler Server, shown in Figure 1. FlashFiler Server is a fully functional and visible application. Another option is FlashFiler Service. FlashFiler Service is an NT service that contains all the functionality of FlashFiler Server, but it's not visible so it's not as easy to change its options. The last option is to create your own FlashFiler server application. You can create a separate server application, like FlashFiler Server, or you can embed the server inside your client application. The basics of setting up FlashFiler Server are covered in the remaining part of this section. The other options are not covered in this article.

Figure 1



Name	State	Clients	Sessions	Open Databases	Open Tables	Cursors	RAM (MB)
Scr	Inactive	0	0	0	0	0	0

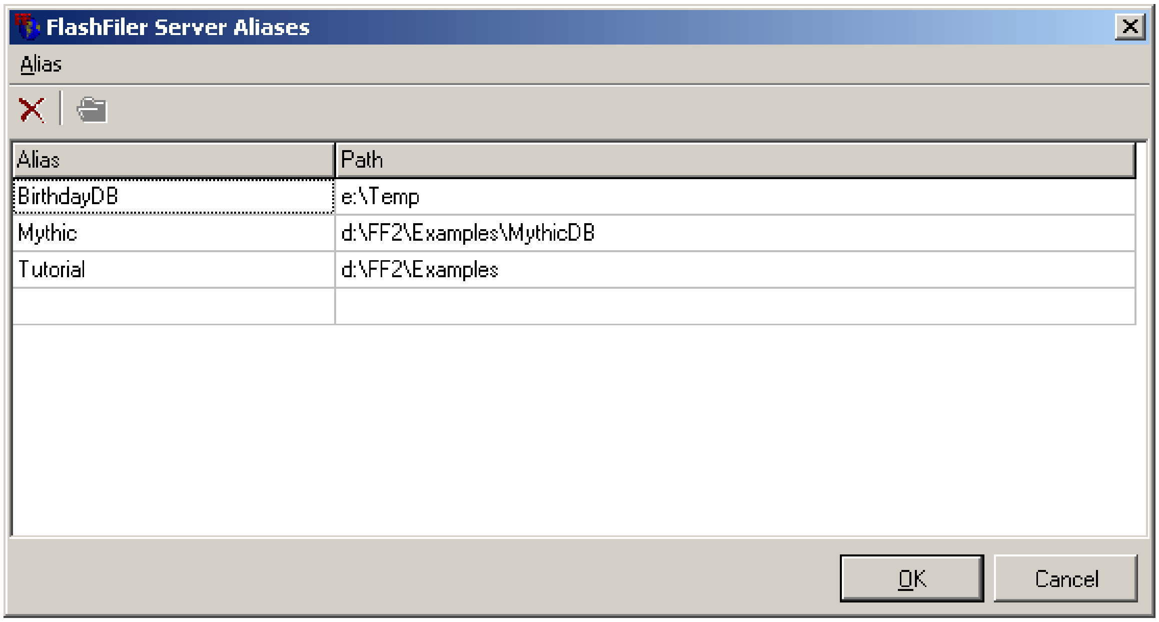
Name	Address	State	Clients	Messages	Messages/Sec
IPX/SPX (FF)	82FD202B:00-00-00-00-00-02	Inactive	0	0	0.0000
Single User (FF)	Local	Inactive	0	0	0.0000
TCP/IP (FF)	192.168.9.131	Inactive	0	0	0.0000

FlashFiler Server's main screen

To prepare FlashFiler Server for its first use, you first need to execute it. Go to FlashFiler's BIN subfolder and execute FFServer.exe. Now, be sure your Server has a name. From the main menu, select Config->General. In the upper left-hand corner of the General Configuration screen, set the Server name to whatever you'd like to call it. That is the only general configuration setting that you need for now, so hit OK to go back to the main screen. Now that you've named the server, you can hit the green arrow on the upper left side of the main screen to start the Server. Starting the server allows it to process requests from FlashFiler client applications.

Before talking about FlashFiler clients, you still need to configure some aliases in the Server. Aliases are short names that identify where database files are located. Clients refer to aliases in order to open tables for a particular database. To configure aliases, go to Config->Aliases… from the main menu. You need to add three aliases. First, add an alias named Tutorial that points to FlashFiler's Examples subdirectory. Add a second one named Mythic that points FlashFiler's Examples\MythicDB directory. Name the third one Birthdays with any location you like. Figure 2 is a sample alias configuration is shown in Figure 2.

Figure 2



Alias	Path
BirthdayDB	e:\Temp
Mythic	d:\FF2\Examples\MythicDB
Tutorial	d:\FF2\Examples

Sample alias configuration for FlashFiler Explorer

The first two aliases allow you to open any of the examples in FlashFiler’s Examples directory. We will use the third one for a small tutorial.

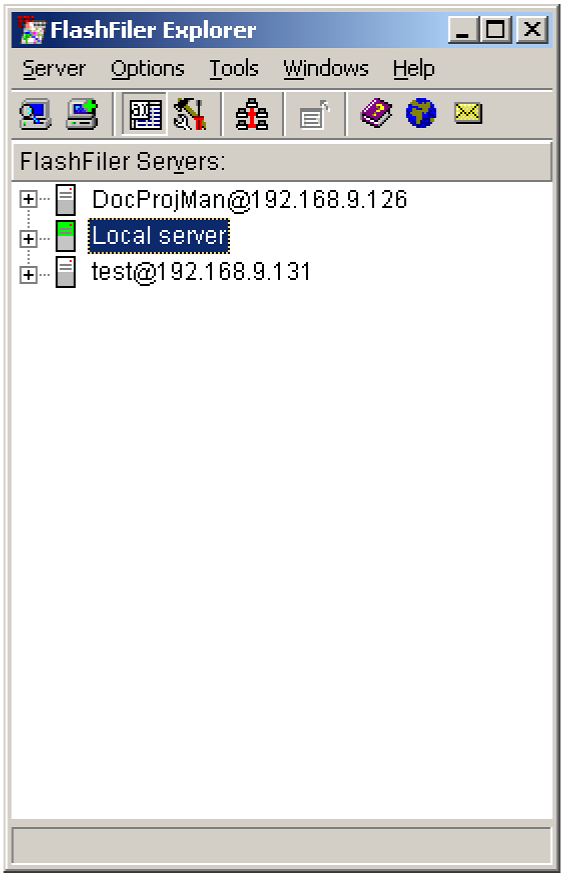
FlashFiler Client Applications

Client applications are what people use to display and modify table. The requests used to retrieve and modify data are virtually transparent to the application developer and end user. As the application developer codes typical database operations, like adding records, filtering data, and querying for data, the database components handle sending and receiving the requests. In other words, switching your applications from a file-sharing system to a client/server system doesn’t require drastic changes.

All of the projects in FlashFiler’s Examples subdirectory are examples of client applications. Feel free to load them into your IDE, start them up, and see how they work. At this point, you should pay attention to how each of the FlashFiler components is used and how they’re connected.

Another client application that is delivered with FlashFiler is FlashFiler Explorer. FlashFiler Explorer allows you to create, view, edit, and delete tables, add, edit, and delete records, and run SQL queries on your databases. It is actually more of a maintenance tool than an application, but it is still a good example of how client applications are used.

Figure 3



FlashFiler Explorer's main screen

Let's startup FlashFiler Explorer and step through some typical tasks. FlashFiler is called FFE.exe and is also located in FlashFiler's BIN subfolder. Go ahead and start it now. Once it's up, it will automatically list all of the FlashFiler servers that it can find. For now, only pay attention to the Local Server. If you expand the Local Server node, you see the databases you added to FlashFiler Server earlier. Under each database is list of tables it contains. Double clicking on a table will let you view the table's data. Feel free to look around and explore the available options before creating a new table.

Now that you're somewhat familiar with FlashFiler Explorer, let's use it to create our first FlashFiler table. This will be a simple table to maintain people's names and birthdays. To create a new table, right click on the Birthday database and select New Table. The screen you're presented with allows you to define a new table. The options here are pretty intuitive. Figure 4 shows how I've defined my table. Define yours the same way, but take time to look at the different options (like the available data types) as you define the table. Once you're finished, click the Create button at the bottom of the screen to tell the server to actually create the new table.

Figure 4

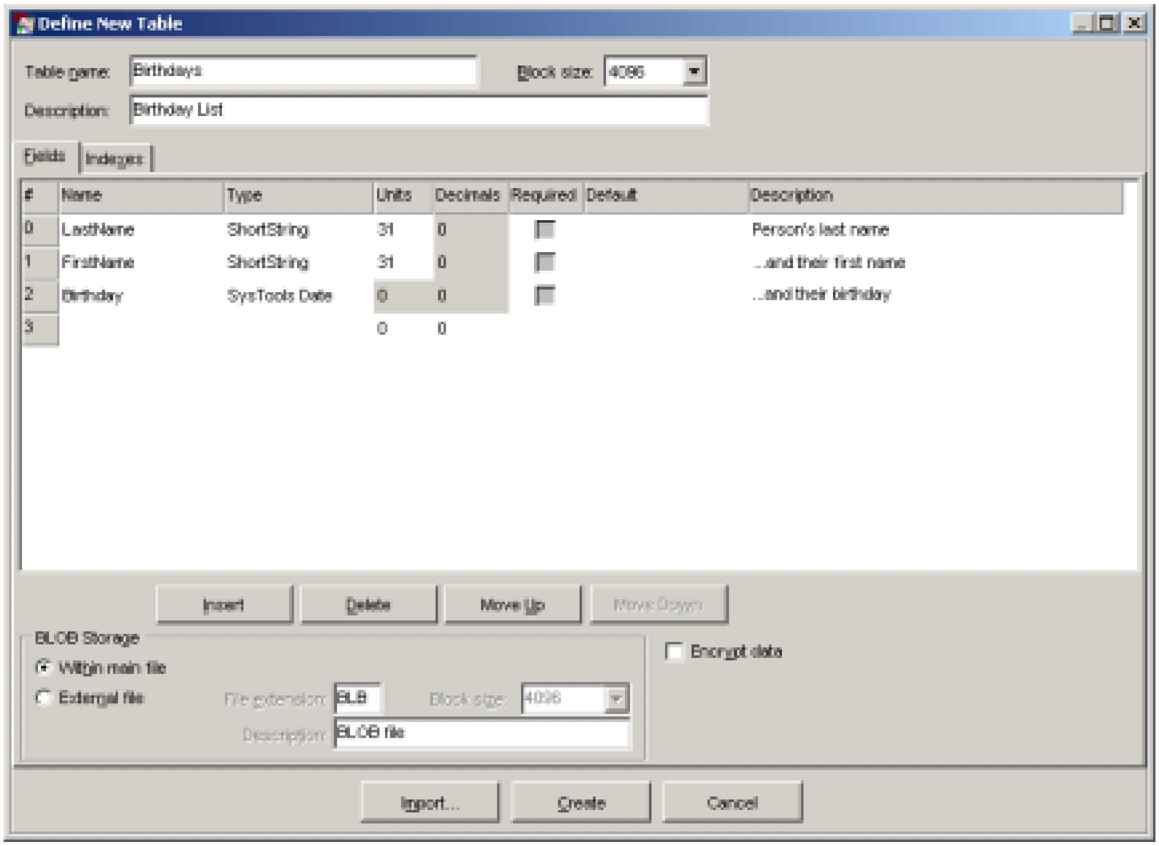


Table name: Birthdays

Block size: 4096

Description: Birthday List

#	Name	Type	Units	Decimals	Required	Default	Description
0	LastName	ShortString	31	0	□		Person's last name
1	FirstName	ShortString	31	0	□		...and their first name
2	Birthday	SysTools Date	0	0	□		...and their birthday
3			0	0			

● Within main file

○ External file

File extension: BLB

Block size: 4096

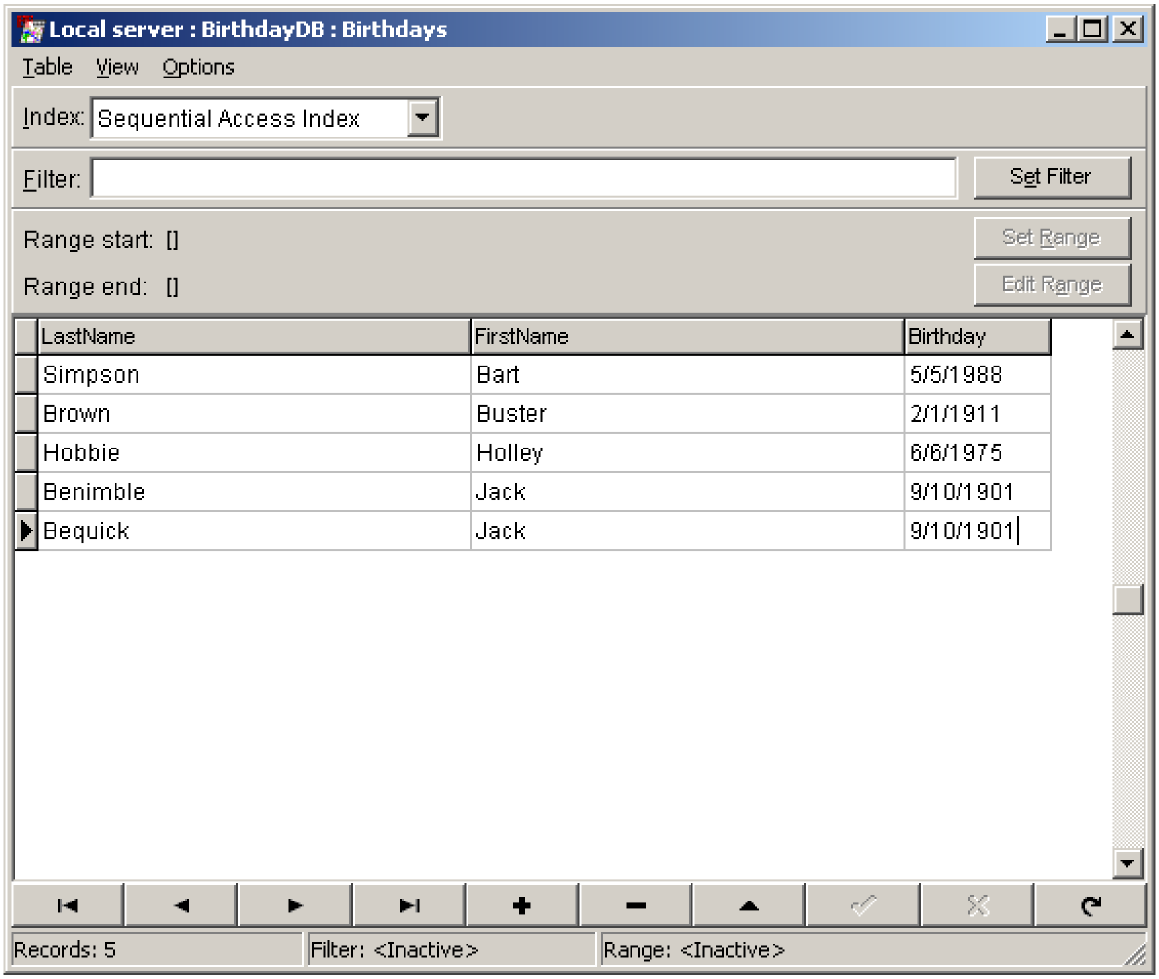
Description: BLOB file

□ Encrypt data

New Birthdays table in FlashFiler Explorer

Now you can use FlashFiler Explorer to add some folks to our new table. Simply double click on the Birthdays table below the Birthday alias. This will bring up the empty table where you can insert, delete, and edit records as I have in Figure 5. Again, feel free to explore the different options on this page. You can simply close this window and FlashFiler Explorer when you're finished exploring.

Figure 5



Index: Sequential Access Index

Filter:

Range start: []

Range end: []

LastName	FirstName	Birthday
Simpson	Bart	5/5/1988
Brown	Buster	2/1/1911
Hobbie	Holley	6/6/1975
Benimble	Jack	9/10/1901
Bequick	Jack	9/10/1901

Records: 5 Filter: <Inactive> Range: <Inactive>

FlashFiler Explorer’s view of Birthdays table with sample data

Want to see how to use the new table in your own FlashFiler client application? It’s a simple process. Here are the steps:

1. Open up your favorite IDE (as long it’s Delphi or C++ Builder <g>).

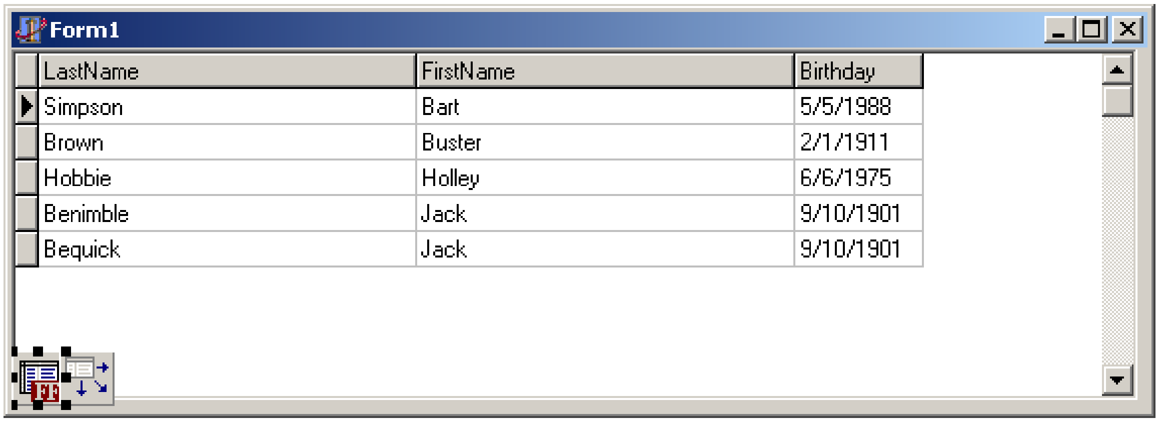
2. Go to the FlashFiler Client component tab and drop a TffTable component on your form. Set its DatabaseName property to Birthday, TableName property to Birthdays, and Active to True.

3. From the Data Access component tab, drop a TDataSource. Set its DataSet property to ffTable1 (this is the default Name property for the TffTable component).

4. From the Data Controls component tab, drop a TDBGrid on the form and set its DataSource property to DataSource1 (this is the default Name property for the TDataSource component). At this point, you can see you data in the grid.

5. Execute the application and you can view, add, edit, and delete records in the table.

Figure 6



LastName	FirstName	Birthday
Simpson	Bart	5/5/1988
Brown	Buster	2/1/1911
Hobbie	Holley	6/6/1975
Benimble	Jack	9/10/1901
Bequick	Jack	9/10/1901

Design-time view of example FlashFiler application

What's Next?

This paper is only a very short introduction to FlashFiler. It really doesn't cover FlashFiler's capabilities or options. For the next step, I highly recommend that you download the **Quick Start Guide for FlashFiler**. Put together by a FlashFiler user, it provides help for creating client applications and shows you how to convert Paradox applications to FlashFiler applications. You can find the Guide under FlashFiler Whitepapers on the TurboPower Foundry at http://community.turbopower.com.

FlashFiler's help file and manual also provide a lot of information and example code. I also highly recommend that everyone visit the FlashFiler newsgroup. The newsgroup is hosted on TurboPower's newsgroup server at news.turbopower.com. The newsgroup is full of very experienced people who are always willing to help out other users. You can also search the newsgroup archives using the Borland section at http://www.fulltextsearch.com.