PRO/5 Automatically Gets 15 years of DBMS Enhancements

With the new Memory Mapped Protocol and !DSOPEN's magic configuration

by Jeff Ash

When a developer determines it is time to migrate an existing PRO/5® application to the latest BBj® platform, gaining access to the plethora of new features is often at the top of the list of reasons for the move. Though BBj is a superset of PRO/5, with over 99% functional compatibility, a common roadblock is the time investment required for thorough testing of application behavior and possible subtle user interface differences between the PRO/5 and BBj interpreter platforms.

BASIS now offers an advantageous intermediate solution. For those seeking to leverage the powerful new BBj DBMS and language features, one can continue to run the application in PRO/5 while transparently invoking the BBj DBMS for file operations and using XCALL to call BBj utilities and functions. The migration is as simple as installing BBj and modifying a configuration file. This is possible thanks to two new features, one in BBj 15.00, and one in PRO/5 15.01. **No data conversion is required** and one can even toggle the functionality on and off during testing! Read on for the background technical details.

Feature One: Avoiding Data Server Syntax and Code Changes

Data server syntax provides a mechanism for instructing PRO/5 to access one or more files on another file server, be it running PRO/5 or BBj. There are very good reasons to use this feature; such as creating multi-tiered environments or providing shared access to the data files from multiple sources. PRO/5 has been able to take advantage of this functionality to access the BBj Filesystem since the first release of BBj 1.0. Applications access data on a data server using a modified file path including a hostname and port to which PRO/5 will connect:

```
    /<myserver,port=2100>/path/to/myfile.dat
```

While the developer has the option to make the modifications in individual file OPEN calls, the most flexible and manageable option is to add configuration settings to the **`config.bbx`** file, such as the PREFIX, making the change for all OPEN calls transparent. With the new PRO/5 15.01 feature however, it's even easier. Simply assign the data server syntax to a special STBL value, !DSOPEN, and there is ***no need to make any modifications to your application code or PREFIX statement***:

```
set !DSOPEN=/<myserver,port=2100>
```

Feature Two: Avoiding the Network Layer and its Overhead

When the primary purpose for the addition of the data server is to use features available only in BBj and there's no need to use multi-tier architecture because the data is on the same server as the PRO/5 interpreter, the new BBj 15.0 Memory Mapped Protocol (MMP) provides tremendous value. For those already familiar with the use of data server syntax, there is virtually nothing new to learn. To use the Memory Mapped Protocol, simply specify 127.0.0.1 for the hostname in the data server path:

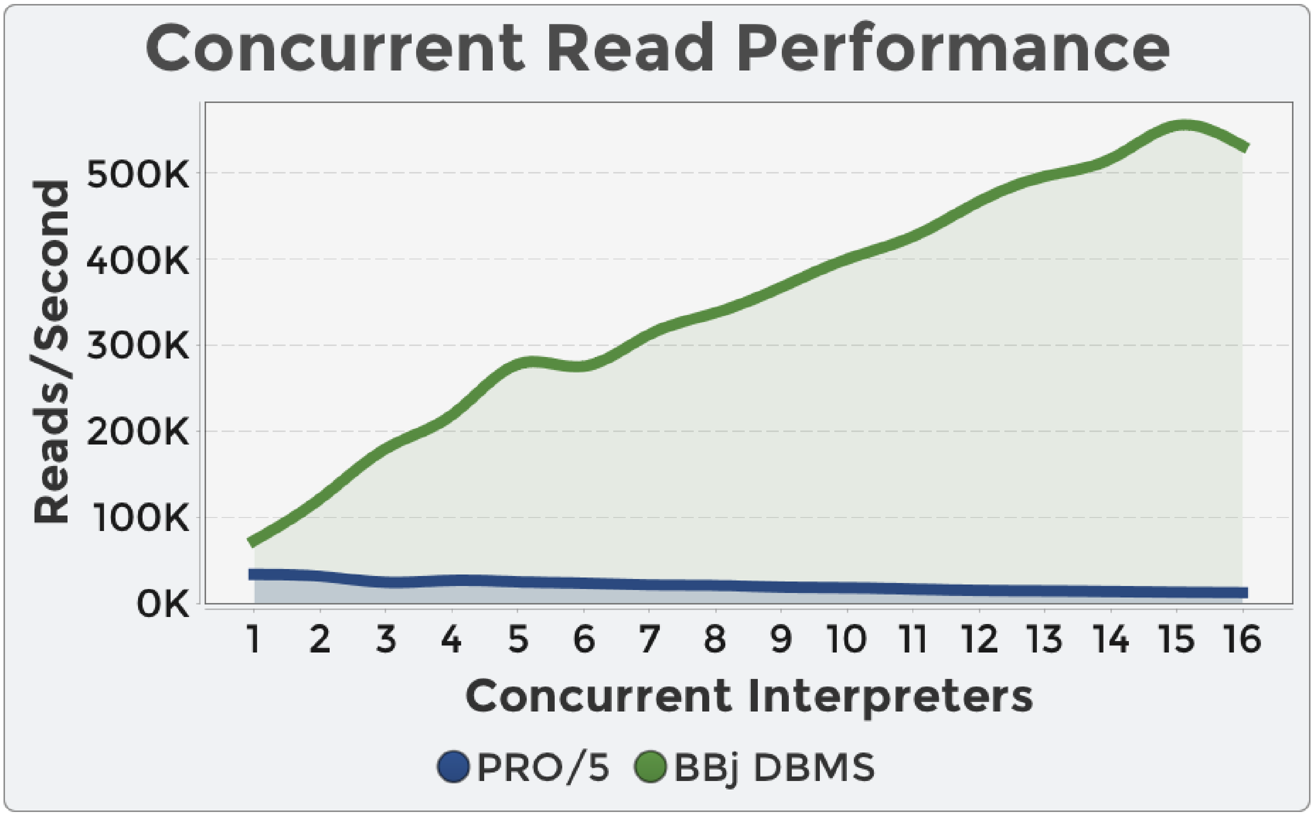
```
/<127.0.0.1,port=2100>/path/to/myfile.dat
```

This instructs PRO/5 to communicate with a running BBjServices PRO/5 DS (Data Server) process via memory instead of the TCP/IP network layer. Bypassing the network layer provides outstanding performance advantages. To implement this feature in conjunction with the !DSOPEN STBL, which obviates the need to change either the PRO/5 PREFIX or PRO/5 application code, simply use 127.0.0.1 as the host name in the DSOPEN! data server path string:

```
set !DSOPEN=/<127.0.0.1,port=2100>
```

Performance Considerations

Performance is always everyone's concern. In measuring performance differences between two different technology architectures it is often tempting to run simple single user "drag race testing", however, this will be wholly misleading when trying to determine real-world experiences of one's multi-user application. A really simple example of the error inherent in this approach can be demonstrated in benchmark testing of single-user versus multi-user data access of a Windows mapped drive configuration versus BASIS data server configuration. In single user testing, the mapped drive configuration outperforms the Data Server configuration by close to 5-fold, but as soon as one adds a second user into the mix, Data Server outperforms mapped drives by more than 30-fold! See the Knowledge Base article KB00713 for more detail. On high user count systems with many concurrent READs, PRO/5 users can expect a significant READ performance boost from BBj's highly efficient algorithms, as can be seen from the chart below:



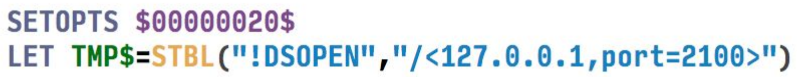
BBj's concurrent READs outperform PRO/5's by up to 10-fold, as the user count grows. In practice, we expect WRITEs to be slower but given that a normal business application performs at least 10x more READs than WRITEs, the net effect should be neutral despite the addition of data communication protocol layer.

Easy to Benchmark
The proof of the pudding is in the eating, and with this new functionality you can move back-and-forth in your testing and evaluate the performance before taking the final plunge. *No data conversion is required.* Batch jobs that may execute a lot of READs and WRITEs in quick succession require examination. However, remember that PRO/5 batch jobs can execute in BBj directly against the data without any communication protocol overhead, and without code recompilation, giving you the best performance for batch jobs. Finally, using the Performance Analyzer to review batch job code invariably leads to substantial performance gains. BASIS offers Professional Services in this regard, and have always been successful in assisting customers to deliver the required performance gains.

Configuring PRO/5 to Use Feature One

Two methods exist for instructing PRO/5 to transparently use data server syntax: (1) programmatically and (2) via the **`config.bbx`** file. The advantage to using the **`config.bbx`** file is that changes to the configuration or path do not require any application code changes. However, as with any change of this magnitude, developers will want to thoroughly test their application with this setting before putting it into production.

Programmatically
To instruct PRO/5 to use a data server syntax prefix for all file OPEN calls, first set the appropriate SETOPTS bit to enable data server communication and follow that with an instruction telling PRO/5 to use a particular data server for OPEN calls:



```
SETOPTS $00000020$
LET TMP$=STBL("!DSOPEN","/<127.0.0.1,port=2100>")
```

Configuration File
Using the **`config.bbx`** method of configuration, developers can easily make the change to use the Memory Mapped Protocol without any application code changes.

Configuration requires two steps:

1. Add or modify the existing SETOPTS line enabling byte 4, bit 20:
   `setopts 00000020`
2. Add the following SET line:
   `set !DSOPEN=/<127.0.0.1,port=2100>`

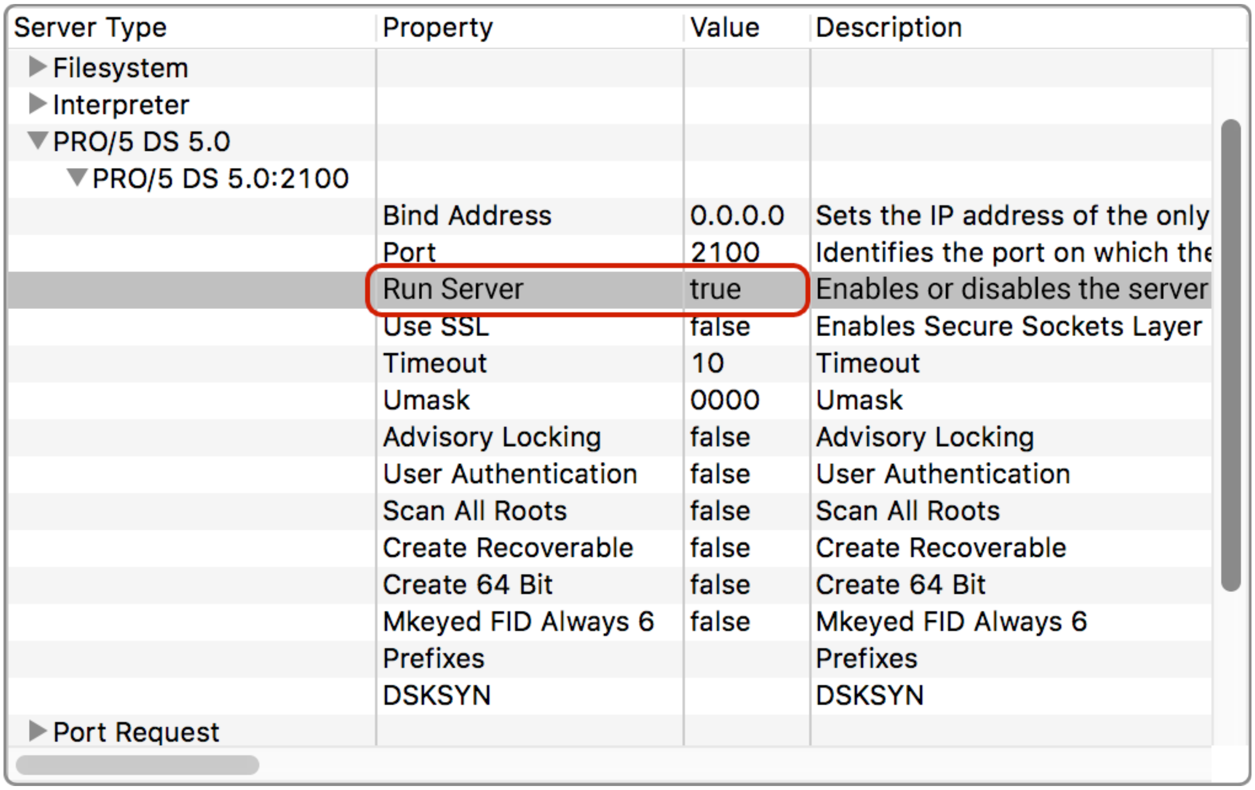
Configuring BBj to Accept PRO/5 Requests and Enable MMP

BBj uses its PRO/5 DS to provide access to the BBj file system from PRO/5 applications. PRO/5 accesses this server either remotely using the hostname of the machine or directly using the Memory Mapped Protocol when PRO/5 and BBj are present on the same machine and PRO/5 uses 127.0.0.1 for the hostname in data server syntax paths.

The following steps describe how to use the Enterprise Manager to configure the BBj PRO/5 DS:

1. **Double click** on the Servers item under BBjServices in the EM navigator to open the configuration page shown in **Figure 1**
2. **Expand** the properties under the PRO/5 DS 5.0 item
3. **Set** Run Server to true to enable the server
4. **Adjust** any additional settings as necessary, such as setting the port number to match the port number in the data server path
5. **Save** the changes

6. **Click** the run server tool button (play icon) or **restart** BBjServices to start the server



Server Type	Property	Value	Description
▶ Filesystem			
▶ Interpreter			
▼ PRO/5 DS 5.0			
▼ PRO/5 DS 5.0:2100			
	Bind Address	0.0.0.0	Sets the IP address of the only
	Port	2100	Identifies the port on which the
	Run Server	true	Enables or disables the server
	Use SSL	false	Enables Secure Sockets Layer
	Timeout	10	Timeout
	Umask	0000	Umask
	Advisory Locking	false	Advisory Locking
	User Authentication	false	User Authentication
	Scan All Roots	false	Scan All Roots
	Create Recoverable	false	Create Recoverable
	Create 64 Bit	false	Create 64 Bit
	Mkeyed FID Always 6	false	Mkeyed FID Always 6
	Prefixes		Prefixes
	DSKSYN		DSKSYN
▶ Port Request			

Figure 1. Configure the BBj PRO/5 DS to run automatically on startup.

BBj Functionality Available When Using Data Server Syntax

With configuration now successfully complete, it is time to take a look at the new functionality available to PRO/5 applications.

Triggers and Asynchronous Triggers

Triggers and Asynchronous Triggers are a powerful and flexible feature consisting of BBj code executed when a particular type of operation occurs on a data file. For example, an after-write trigger executes its associated BBj code after a write occurs on the data file. A before-read trigger executes its code before a read occurs on the data file.

Data Replication

BBj provides a robust replication framework useful for creating data backups, data warehouses, and to offload reporting activity to an alternate server. Using the Enterprise Manager, the administrator configures any number of replication jobs to replicate anything from individual data and program files to entire databases and sets of directories.

Change Auditing

When business requirements include the ability to track changes made to one or more data files or even the entire database, change auditing meets this need. Using the Enterprise Manager, the administrator creates audit jobs that track all monitored data files for changes and logs those changes to the audit database. The administrator queries this data using the helpful user interface to locate specific changes, including the time of the change, old and new values, and who made the change.

Other Features

In addition to the first three features which are directly filesystem related, installing BBj opens up a world of other powerful capabilities such as task scheduling, improved ODBC functionality, JDBC access, document indexing, FULLTEXT searching in SQL, provisioning Browser User Interface (BUI) applications, implementing Web Services, adding Business Intelligence (BI) with the Dashboard Utility and much more! With XCALL, all backend BBj functionality, including utilities like the email utility are available to your PRO/5 application.

Summary

Ultimately, every PRO/5 developer benefits from migrating their applications to BBj. However, this process may require more time than the current day to day operations allow. Fortunately, an incremental move to BBj may be a viable option. Begin by migrating data access to the BBj DBMS via the BBj PRO/5 Data Server with a simple configuration change. The developer continues to use PRO/5 for the application interpreter but opens up a world of possibilities to begin leveraging the powerful features available in the BBj DBMS and the BBj language. Using the new BBj Memory Mapped Protocol requires just a few changes and provides PRO/5 users with the ability to incorporate some of the powerful features found in BBj without any additional network overhead and without changing any application code.

Download the latest versions of BBj and PRO/5 today to try it out for yourself! Contact your Account Manager if you need further assistance or to setup a technical call to answer any questions that you may have about your particular situation or your specific needs that this new solution will be able to meet.