

Caché *and* **MultiValue:**

Birds of a Feather – Part 2

BY TONY GRAVAGNO

It's possible that some new or existing Caché developers will start using MV BASIC as just another language in their box of tools.

In the first article of this series, I described the Caché DBMS in general terms, introduced InterSystems as the company that provides Caché, and explained what various people in our market stand to gain now that Caché supports MultiValue technology. In this article I will provide more insight into the software and the company, and then shift from that high level view to describe some of the technical aspects of their MultiValue implementation.

What is Caché?

As mentioned in Part 1, Caché (pronounced Cashay) is very much like its MultiValue DBMS counterparts. However, SQL integration with Caché is much tighter than with most MultiValue platforms which add various degrees of SQL support via dictionary mappings and other methods. Caché is famous for being an object database, which means it natively supports complex data objects in a way that flat relational databases cannot. While the MultiValue data model can be considered to be somewhat object-oriented, Caché supports objects to its core, and

much deeper than the item, attribute, value, sub-value structures that we enjoy. I will provide more detail about the data model with examples in an upcoming article.

Among other distinctions, Caché has been recognized as being used in 16 of the Healthcare Informatics Top 20 Healthcare IT Companies for 2006, and by all 14 of the Best Hospitals from a 2006 US News Honor Roll. Caché has received the prestigious Common Criteria certification, an international security standard recognized by 24 countries. And let this one sink in for a minute—there is a com-

pany mandate that every release of Caché must be faster than its predecessor, even as new features are added.

InterSystems

As mentioned in the first article, InterSystems is not offering us just another MultiValue database. Their database and other technical components are all excellent and worthy of consideration, but to a limited extent most of the current MultiValue DBMS products have similar technical capabilities which will be discussed in another article. Aside from the technical offering, I think the real value-add from InterSystems is their own business. Unlike MultiValue DBMS companies (rather, “other MultiValue DBMS companies”), they are well known in the IT industry, in governmental agencies, and among mainstream business and IT managers, and they are promoting “our” technology. One could correctly cite IBM as being well known, but in all these years that hasn’t yet translated into recognition of MultiValue technology in the mainstream. Full-page ads in InformationWeek magazine now include a note that Caché now supports MultiValue development. We can expect more of these ads in the future since MultiValue is now as integral a feature as any other, and InterSystems isn’t afraid to advertise it.

It may help to know that this technical and business initiative for MultiValue is supported by many people with extensive MultiValue backgrounds. I’ve met several of their development engineers and Sales Engineers (SE’s). They’re intimately familiar with MultiValue because they’ve been in the MultiValue industry for (as we often say) “longer than we sometimes care to admit”. Their Product Manager, QA Manager, and their lead Sales representative for MV all have extensive MultiValue backgrounds.

In my next article I’ll comment more on the management and culture at InterSystems. Regardless of the business these people are in, I think their organization is an excellent model for other companies, and worthy of recognition.

BASIC

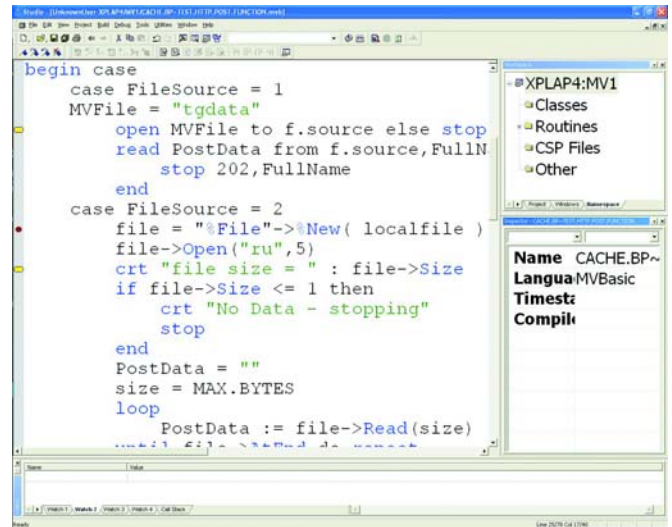
Let’s shift now to discuss a technical aspect of the environment which is critical and of interest to all. The two languages traditionally used by Caché developers are Caché Object Script (COS), and their own version of BASIC which was added for those porting applications written in Visual Basic or VB

Script. Multivalue BASIC is now yet another language available to all Caché developers. It is not an add-on and not just for migrations. MV BASIC is what I call a real “first class citizen,” integrated into the core product and available for use wherever code can be used, including class definitions, web pages, and web services. This is possible because all Caché languages compile down to the same object code (p-code is a term familiar to many MultiValue people). So everyone uses the same run-time engine regardless of their choice of languages.

Of course MV BASIC will be used primarily by sites migrating from MultiValue-based environments, but not exclusively. The simplicity and efficiency of the language, enjoyed by MultiValue developers for decades, is now available to all Caché developers, and it’s possible if not likely that some new or existing Caché developers will start using MV BASIC as just another language in their box of tools. For these reasons, the language will not be deprecated, developers will not be asked to switch to other Caché languages later, and support for MV BASIC will not diminish after InterSystems gets some number of MultiValue migrations.

As expected, the MV BASIC implementation supports most common syntax and many applications compile and run with virtually no issues. To say it already supports absolutely all statements, functions, and other language

Figure 1



elements would be an exaggeration, but some sites find that all of their code compiles on the first pass. Some sites are finding that code which does not compile isn’t used anyway. We frequently have old programs that were tests, never used, replaced, etc. Some of these may not even compile on our source systems, let alone in a new environment, and sites are finding these relics when they attempt to compile all programs. After compilation, both local and global catalogues are supported just like any MultiValue environment with this feature.

The BASIC runtime also behaves as MultiValue developers would expect, though even here, some platform-specific code may need a second look. As an example, I’m working with Pete Schellenbach to port AccuTerm to Caché. When we used the UniVerse code base, everything compiled, but there was one run-time error. This disappeared when we removed some code that was originally added just to accommodate some unusual UniVerse behavior. After that one change, I could enable another account with AccuTerm, and then logto that account and use the WED GUI to edit and compile programs. When I used the QM code base there were no issues at all. There’s other code that Pete says needs to be tweaked for every new platform—I’ll let you know what happens with that in the next article.

Speaking of GUI, Caché Studio (shown in Figure 1) is used for development with Caché Object Script, Caché BASIC, and MultiValue BASIC. (You can use ED for all of this too, but what fun is that?) Studio is used to edit Caché Server Pages (like Microsoft Active Server Pages), XML documents, JavaScript, Cascading Style Sheets, Web Services, and code related to the new Zen browser UI development tools. All of this is free and built-in to the default Caché environment, and MV BASIC can be used wherever code is required.

To debug a program we don't one-step through asterisk prompts in a telnet window. We run a program from a telnet window, then attach to it from Studio, then one-step in Studio in a manner very familiar to anyone who has used Microsoft Visual Studio. One will find that the runtime error messages from BASIC programs can be very cryptic until one understands exactly what they're trying to say. Debugging is indeed different at first, but quickly becomes more natural.

All in all, MultiValue developers should not have a problem working with BASIC in Caché.

Emulations

Sites running UniVerse or UniData are quite familiar with "flavors". D3 developers have \$OPTIONS in code which allows code to run with GA, Ultimate, and other platform-specific syntax. QM and other products have options and switches that allow code to compile and run in a way which emulates the behavior of other MultiValue DBMS platforms.

Caché also supports emulations, 16 so far, allowing many applications to be ported into this environment with few or no changes. The most developed emulations are for UniVerse and jBASE. This should be no surprise as there are certainly a large number of UniVerse sites out there, and most of the developers working on the MultiValue enhancements at InterSystems have a strong background with jBASE.

Since one of the primary emulations is for UniVerse, accounts are initialized with verbs familiar to UniVerse users but less familiar for those running a DBMS like Reality or D3. For example, there is a LISTF verb, but no LISTFILES or LIST-FILES. Verbs and file pointers are contained in a VOC, which is familiar to U2 and Prime-flavor users but largely unknown to many users on the Pick side. For Pick users (D3, AP, mvBASE, etc) there is an item in the VOC called MD, pointing recursively back to the VOC, thus allowing for statements like "SORT MD". Verbs and other MD/VOC items can be added manually, and InterSystems will process requests for enhancements to support common verbs which have not yet been implemented.

Summary

As you can see, so far this is as MultiValue as any other DBMS in our market. The process of evaluation and/or migration includes porting code, dictionaries, and data; investigating failures; eliminating what's not necessary; fixing what's easy; and reporting the remainder to InterSystems as a request for a fix or enhancement. In this respect it's the same as other MV-MV migrations. But there are added bonuses: The environment is as much relational as it is MultiValue. It's already supports data objects and development when other MultiValue environments are implementing or just thinking about the concept of object-oriented code. And once an application has been ported to Caché, it should have much more marketability to the outside world, as well as more credibility with internal management.

We like our Pick / Prime / MultiValue software for many reasons, including ease of development and maintenance, and to some extent we also enjoy a lower cost of ownership compared to many sites running other DBMS platforms. But unless this market grows and becomes better recognized for its strengths, many of us will continue to have problems advocating this soft-

ware to management and new sales prospects. Now that Caché supports MultiValue applications, we can get the same business benefits as other Caché users and resellers, and still enjoy the MultiValue-specific benefits that we value so highly. I think we have a significant new opportunity here, and I hope these articles will help to convey exactly why I feel this is such a good fit for many sites that are still looking for that "right" solution.

In my next article I will spend more time explaining the Caché data model, emulations, and other technical details of interest to MultiValue developers and administrators. Thank you for your time and interest. IS



TONY GRAVAGNO

is President of Nebula

Research and Development, providing worldwide consultation and advanced development services for Pick/MV VARs and end-users. To discuss Caché, training, migration services, or for more information about anything discussed here, please visit:

<http://Nebula-RnD.com/spectrum>

INTERSYSTEMS

www.InterSystems.com