

Clarion Magazine

Reborn Free



Greenbar By The Block

Henry Plotkin shows a neat trick for adding a greenbar type effect to an event log.

Posted Thursday, September 09, 2004

Converting Clarion 5.5 Apps To Clarion 6.1

Dermot Herron had no intention of upgrading from C55 to C6.1, but circumstances forced him to confront his concerns over the huge step to true threading. Unexpectedly, he found that it's easy to make a C6.x application work just like a C55 application, and that means he can pursue the full conversion at a later date while reaping many C6.x benefits now.

Posted Thursday, September 09, 2004

Product Review: The Ripley Code Commentor

The Ripley Code Commentor, by Scott Ripley, is a nifty new utility that makes commenting code changes fast and easy, not just for Clarion but for other languages/development environments as well.

Posted Friday, September 10, 2004

DevCon 2004 Reporters Wanted - Three Sessions and a Wrap-Up

We still need just a few more reporters

Articles: 

News: 



News

[UK Fenix Training](#)

[Clarion Developers Challenge Announcement](#)

[BoxSoft Super Stuff Status](#)

[vuFileTools Demo](#)

[Solace Software Releases New Live Update Templates](#)

[ADDA 1.0.4](#)

[RUpdate 1.b Beta Released](#)

[Clarion Developers Challenge Week 2 Winner](#)

[ThinkData 30% Off Sale](#)

[Clarion Developers Challenge Week 1 Winner Announced](#)

[CapeSoft Support Notice](#)

[SelfService Goes Gold](#)

[New HotDates From CapeSoft](#)

[New GPFReporter From CapeSoft](#)

[New AnyFont From CapeSoft](#)

[Free Sybase ASE Express](#)

SURVEY

The . shorthand for END will disappear in Clarion.NET. I am:

Horrified
 16.5%

Disappointed
 13.5%

Indifferent
 37.5%

Delighted
 32.5%

200 responses

[Previous Surveys](#)

One Year Ago In CM

[Weekly PDF For September 14-20, 2003](#)

[PDF For September, 2003](#)

[Product Review: NetTalk, Part 1](#)

for DevCon 2004 - there are three sessions and the conference wrap-up to be reported on. As there are a lot of sessions, we're not asking for really long reports (although you can always go for it if you want to).

Posted Thursday, September 16, 2004

Break Dancing With Clarion

Clarion 6.1 has introduced some improvements to the report designer. These aren't the sort of changes that will, on their own, make you want to run out and upgrade to C6.1, but they are important advances nevertheless. In the first article in this review Geoff Bomford covers the use of pre-Clarion 6.1 techniques for performing some common report break tasks, and in the next article will discuss the use of the BreakMgr Class, a new tool in Clarion 6.1 for handling the same tasks a little better!

Posted Thursday, September 16, 2004

Dynamic Table-Based Tagging

Sometimes a simple tagging solution is the ideal solution, and sometimes you need something a bit more robust. When his users began requiring features not included in a readily available tagging class, Chris set out to create his own tagging functionality using a highly flexible, table-based approach.

Posted Friday, September 17, 2004

Handling Nullable Fields in an ABC Application

It's no secret that Clarion doesn't

[Edition For Linux](#)

[ClarionMag DevCon Sale](#)

[DeveloperPLUS Store
Temporarily Offline](#)

[Clarion Developers
Challenge Vendor of the
Week](#)

[PowerOffice DevCon
Specials](#)

[OutlookBar 1.2](#)

[Clarion Developers
Challenge Now Open](#)

[Updated CMag Automatic
Logon Utility](#)

[Clarion Third Party Profile
Exchange Updated](#)

[New gReg Site](#)

[Comparison Between the
Two Approaches to
Achieve Net-Centricity](#)

[Search the news
archive](#)

Two Years Ago In CM

[The Clarion
Challenge -
ContainsMatch](#)

[How To Ignore A
Template](#)

[Data Structures
And Algorithms
Part IX - Are
You Getting Too
Tall?](#)

Three Years Ago In CM

[Review:
LogFlash 2.4
from Sterling
Data](#)

[Clarion News -
October 2001](#)

[PDF for
September 23-
29, 2001](#)

Four Years Ago In CM

[Five Rules for
Managing
Complexity: Part
4](#)

[Five Rules for
Managing
Complexity: Part
5](#)

[Clarion News -
September 2000](#)

handle nullable fields very well. In keeping with the ideal of simplicity, Clarion considers the value of a Null string to be blank, and a Null number to be zero. And in most cases, this is fine, and can make your code easier to use and maintain. But there are occasions when you have a Nullable field that really needs to be Null.

Posted Friday, September 17, 2004

DevCon 2004 Starts Off In Style

DevCon kicked off Sunday with a reception at the Disney Contemporary Resort marina, under a beautiful Florida sky, with cooler than expected temperatures (but still plenty warm for those of us not used to the local climate). Good food and drink, and a chance to catch up with old friends and make new ones.

Posted Sunday, September 19, 2004

DevCon 2004 Monday Keynote: Bob Zaunere (Part 1)

Bob Zaunere, SoftVelocity's president, gave the keynote address on Monday morning. Bob Z set the promised conference tone with a complete focus on Clarion technology, and a total absence of "marketing hype." He covered some upcoming new C6.1 products, including three new drivers, and then spent some time talking about Clarion 7's feature set. Part 1. (This is a free access article)

Posted Monday, September 20, 2004

DevCon 2004 Monday Keynote: Bob Zaunere (Part 2)

Bob Zaubere, SoftVelocity's president, gave the keynote address on Monday morning. Besides discussing a number of C6.1-compatible new products, he discussed the Clarion 7 feature set. This article also has a link to some video from Bob's talk. Part 2.

Posted Tuesday, September 21, 2004

DevCon 2004 Keynote: **Clarion.NET**

Tuesday's keynote address by Bob Zaubere was titled "From Win32 to .NET", and gave the attendees a look into the not-very-distant future. And the future is Clarion.NET. (Free access article)

Posted Tuesday, September 21, 2004

DevCon Pre-Conference **Training: Day 1 & 2**

With more than 140 developers in attendance this wasn't your usual training class. In fact it was more of a guided tour of the new features of Clarion 6 than training. Greg Bailey reports.

Posted Thursday, September 23, 2004

DevCon 2004: Developing **Applications with the new IP,** **Memory and Dynamic File** **Drivers**

This session was delivered by the inimitable Bob Foreman of SoftVelocity, and was intended as an introduction to three new and/or recently announced add-ons for Clarion.

Posted Thursday, September 23, 2004

DevCon 2004: Clarion In The Public Sector

This presentation by Douglass Smith may have been the sleeper hit of DevCon 2004. Steve Gradolph reports.

Posted Wednesday, September 29, 2004

DevCon 2004: Using Objects in Clarion

For those of you who have never heard Bruce Johnson explain OOP and virtual methods, you have missed one of the truly enlightening experiences for a Clarion programmer.

Posted Wednesday, September 29, 2004

DevCon 2004: COM Objects In 15 Minutes Or Less

The longest presentation at DevCon was Jim Kane's two-part session on creating a COM object using Clarion. While this is still not a simple task, Jim's free utility makes the task possible with a minimum of pain.

Posted Thursday, September 30, 2004

DevCon 2004: Debugging Clarion Apps

Russ Eggen gave a large crowd at DevCon a tour of the Clarion debugger, and provided a number of valuable debugging tips.

Posted Thursday, September 30, 2004

DevCon 2004: The PHP Templates

Pierre Tremblay demonstrated SoftVelocity's new PHP template set, which is expected to be available in about a month.

Posted Thursday, September 30, 2004

DevCon Pre-Conference

Training: Day 3

While Day 1 and 2 of the pre-conference training sessions focused on new functionality provided by C6, Day 3 focused on taking advantage of capabilities available to us since Clarion 4 introduced Object Oriented Programming (OOP) and the Application Building Classes (ABC).

Posted Thursday, September 30, 2004

Looking for more? Check out the [site index](#), or [search the back issues](#).

This site now contains more than 700 articles and a total of over a million words of Clarion-related information.

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)



Greenbar By The Block

by Henry Plotkin

Published 2004-09-09

In a point of sale system I work on, transactions are stored in a log file. These logs are used to update the inventory and sales files. It was important for me to be able to view these logs. TopScan could work but the layout did not quite fit my needs. So I created a log viewer for myself. Figure 1 shows my log viewer.

My Place For log for 9/03/04

Records in this log: 25

Store ID: 1 Receipt Number: 808 Cashier:
 Register: 1 Receipt Date: 9/03/04 Sales Clerk:

Show highlighted receipt only

Rec't #	Rec't Time	Trans	PLU	Quantity	Ext. (Unit Price)	Ext. (Net of Dept)	Unit Cost	Sale Tax Fl
808	7:56AM	1	1000	1.000	1.00	1.00	2	0.25 S 1
809	7:57AM	1	1000	1.000	1.00	0.90	2	0.25 S 1
809	7:57AM	21	10% Discount	(1.000)	0.10	0.10	2	0.25 P 1
810	8:00AM	1	1000	1.000	1.00	0.75	2	0.25 S 1
810	8:00AM	19	\$0.25 discount	(1.000)	0.25	0.25	2	0.25 D 1
811	8:01AM	1	1000	1.000	1.00	0.90	2	0.25 T 1
811	8:01AM	22	10% Discount	(1.000)	0.10	0.10	2	0.25 P 1
812	8:03AM	1	1000	1.000	1.00	0.75	2	0.25 T 1
812	8:03AM	20	\$0.25 discount	(1.000)	0.25	0.25	2	0.25 D 1
813	8:04AM	1	1000	1.000	1.00	0.90	2	0.25 S 1
813	8:04AM	136	10.00% discount	(1.000)	0.10	0.10	2	0.25 c 1
814	8:16AM	1	1001	1.000	2.00	2.00	1	0.50 1
815	8:18AM	1	1000	1.000	1.00	1.00	2	0.25 S 1
815	8:18AM	1	1000	1.000	1.00	1.00	2	0.25 S 1
815	8:18AM	1	1002	1.000	3.00	3.00	1	1.00 1
816	8:19AM	1	1001	1.000	2.00	2.00	1	0.50 1
817	8:20AM	1	1000	2.000	2.00	2.00	2	0.25 S 1
817	8:20AM	1	1002	1.000	3.00	3.00	1	1.00 1
818	8:21AM	1	1001	1.000	2.00	2.00	1	0.50 1
819	8:22AM	1	1000	1.000	1.00	1.00	2	0.25 S 1
819	8:22AM	1	1000	1.000	1.00	1.00	2	0.25 S 1
819	8:22AM	1	1002	1.000	3.00	3.00	1	1.00 1
820	8:24AM	1	1001	1.000	2.00	1.80	1	0.50 1
820	8:24AM	136	10.00% discount	(1.000)	0.20	0.20	1	0.50 c 1
821	8:25AM	1	1002	1.000	3.00	3.00	1	1.00 1

Customer Number:

Figure 1. Log view utility ([click here](#) for a full-sized image)

This view does organize material as I would like but I cannot tell one transaction from another just by looking or when scrolling. Figure 2 shows alternating transactions with different backgrounds This allows me to visually tell one transaction from another.

My Place For log for 9/03/04

Records in this log: 25

Store ID: 1 Receipt Number: 808 Cashier:

Register: 1 Receipt Date: 9/03/04 Sales Clerk:

Iran Code Descriptions
Print Change Log

Show highlighted receipt only

Rec't #	Rec't Time	Trans	PLU	Quantity	Ext (Unit Pri)	Ext (Net of Dept)	Unit Cost	Sale Tax Fl
808	7:56AM	1	1000	1,000	1.00	1.00	2	0.25 S 1
809	7:57AM	1	1000	1,000	1.00	0.90	2	0.25 S 1
809	7:57AM	21	10% Discount	(1,000)	0.10	0.10	2	0.25 P 1
810	8:00AM	1	1000	1,000	1.00	0.75	2	0.25 S 1
810	8:00AM	19	\$0.25 discount	(1,000)	0.25	0.25	2	0.25 D 1
811	8:01AM	1	1000	1,000	1.00	0.90	2	0.25 T 1
811	8:01AM	22	10% Discount	(1,000)	0.10	0.10	2	0.25 P 1
812	8:03AM	1	1000	1,000	1.00	0.75	2	0.25 T 1
812	8:03AM	20	\$0.25 discount	(1,000)	0.25	0.25	2	0.25 D 1
813	8:04AM	1	1000	1,000	1.00	0.90	2	0.25 S 1
813	8:04AM	136	10.00% discount	(1,000)	0.10	0.10	2	0.25 c 1
814	8:16AM	1	1001	1,000	2.00	2.00	1	0.50 1
815	8:18AM	1	1000	1,000	1.00	1.00	2	0.25 S 1
815	8:18AM	1	1000	1,000	1.00	1.00	2	0.25 S 1
815	8:18AM	1	1002	1,000	3.00	3.00	1	1.00 1
816	8:19AM	1	1001	1,000	2.00	2.00	1	0.50 1
817	8:20AM	1	1000	2,000	2.00	2.00	2	0.25 S 1
817	8:20AM	1	1002	1,000	3.00	3.00	1	1.00 1
818	8:21AM	1	1001	1,000	2.00	2.00	1	0.50 1
819	8:22AM	1	1000	1,000	1.00	1.00	2	0.25 S 1
819	8:22AM	1	1000	1,000	1.00	1.00	2	0.25 S 1
819	8:22AM	1	1002	1,000	3.00	3.00	1	1.00 1
820	8:24AM	1	1001	1,000	2.00	1.80	1	0.50 1
820	8:24AM	136	10.00% discount	(1,000)	0.20	0.20	1	0.50 c 1
821	8:25AM	1	1002	1,000	3.00	3.00	1	1.00 1

Net Subtotal: 1.00

Taxes: Tax 1: 0.08
Tax 2:
Tax 3:
Tax 4:

Total Tax: 0.08
Total: 1.08

Tenders: Cash: 2.00
Check:
Charge 1:
Charge 2:
Hse Chg:
Food Stamps:
Credit Card:
Store Cr:
Cust Cr:
Lay Away:
Gift Card:
Gift Cert:
Coupon:

Change: 0.92

Customer Number:

View Close

Figure 2. Transactions highlighted ([click here](#) for a full-sized image)

This is similar to the green bar capability already part of Clarion. But, instead of changing colour on each row, it changes when the transaction ID changes.

Green barring an entire block of rows is easy. First, in the List Box Formatter, set the color attribute (see Figure 3).

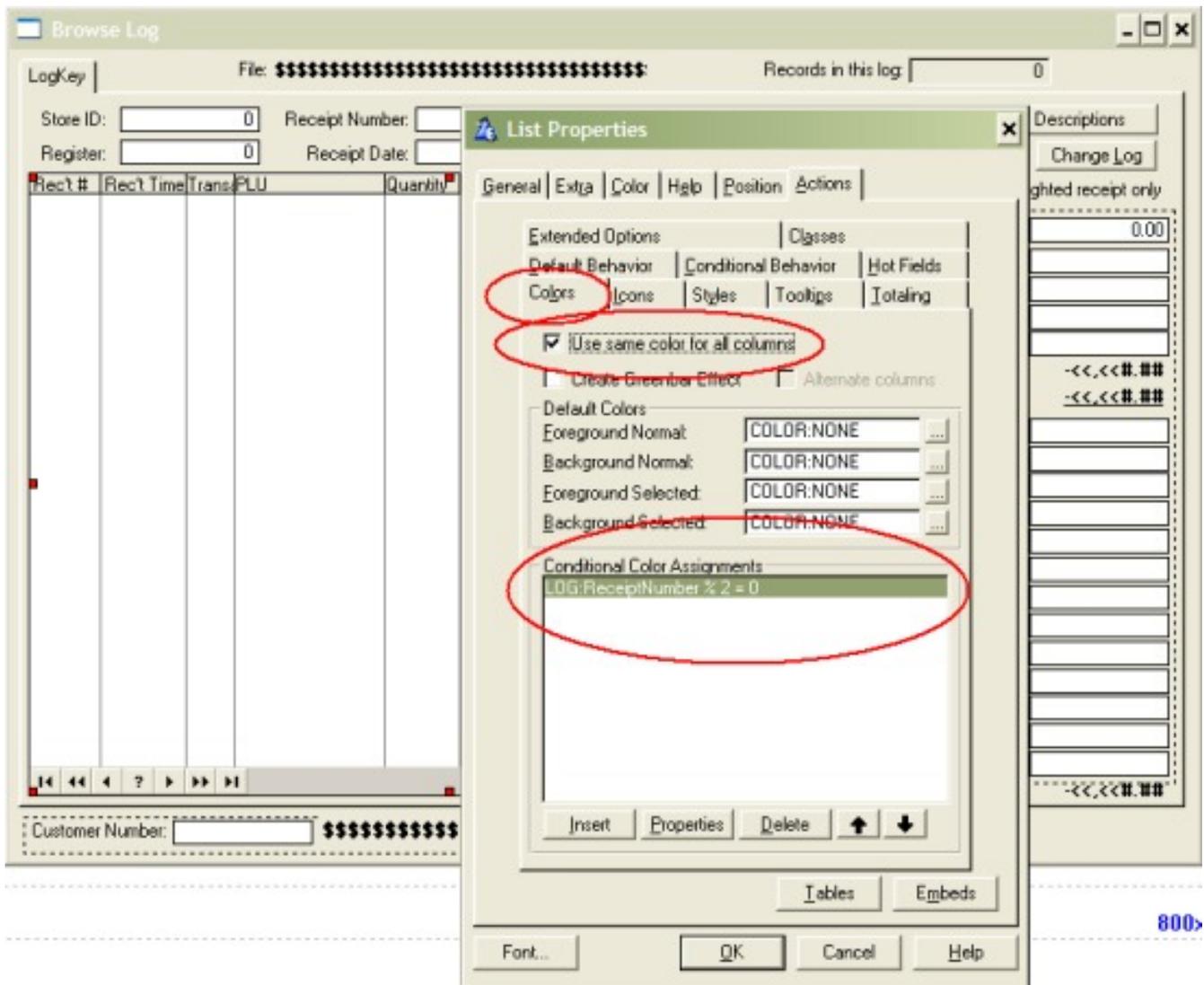


Figure 4. Setting all columns ([click here](#) for a full-sized image)

Press "Properties" to create a condition which will trigger the colours.

Since transactions are numbered sequentially (autonumber), LOG:Receipt % 2 distinguishes odd from even numbers and changes the colour on even numbered transactions.

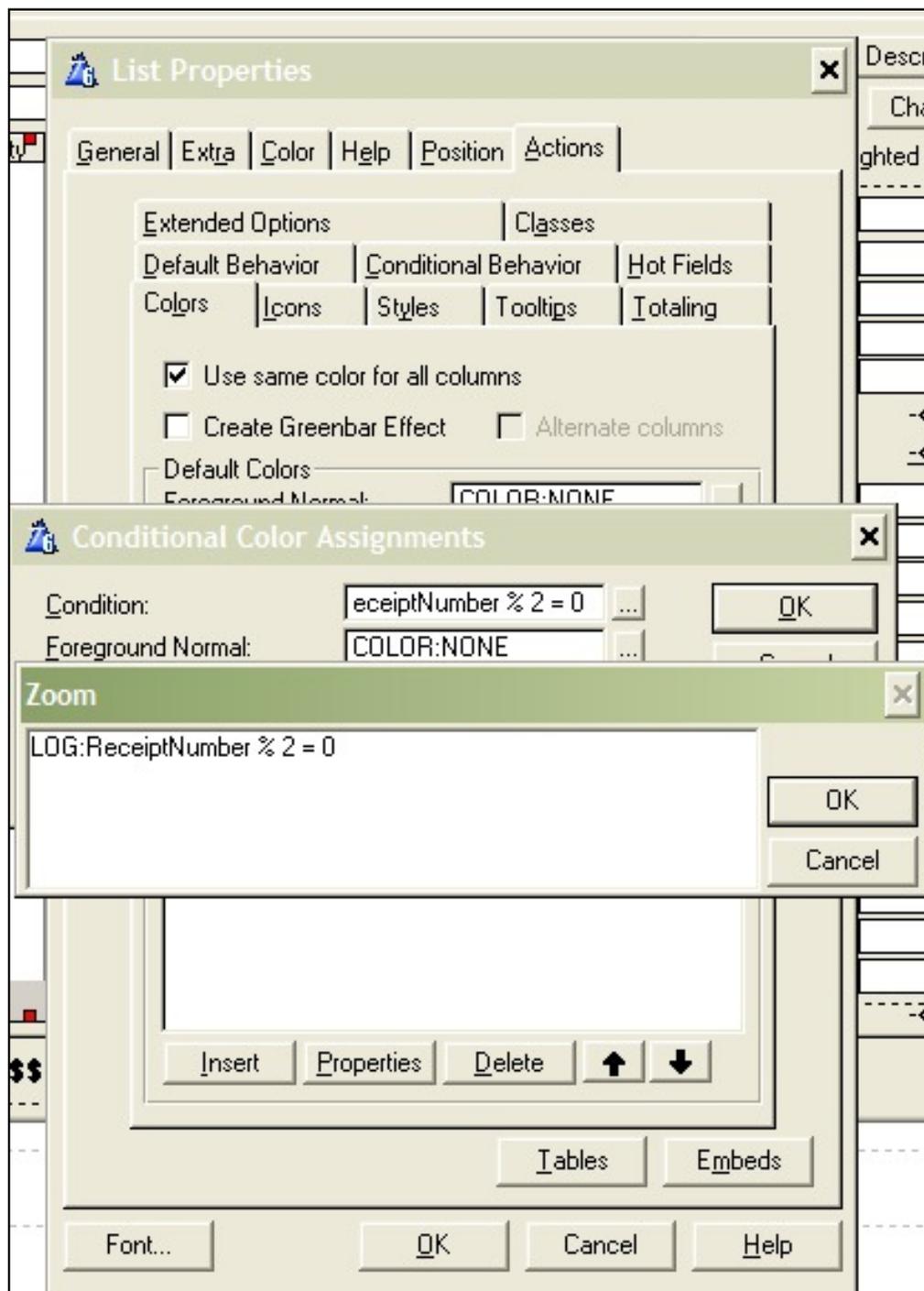


Figure 5. The condition

The result is seen in Figure 2, earlier, and is just what I need.

"hp" in fact prefers Hewlett-Packard printers but will use whatever is available. Born in New York City, hp is a self-taught Clarion developer doing a substantial amount of work for hospital gift shops.

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)



Converting Clarion 5.5 Apps To Clarion 6.1

by Dermot Herron

Published 2004-09-09

Clarion 6.1 is a huge step forward, wonderful and immensely powerful, highly desirable, nearly bug-free, *very* stable and reliable and, so it seemed to me, impossibly complex with its pre-emptive multitasking.

I think the folks at SoftVelocity have made a basic mistake in emphasis in Clarion 6x. Pre-emptive multitasking is technically *very* clever and *very* necessary and very important under-the-hood, but in my opinion C6 should have come out automatically *defaulting* to cooperative multitasking when importing a C55 app (and maybe always), with the *option* of turning on pre-emptive multitasking! The conversion rate from C55 would have been very much higher. Backward compatibility is so important that it *must* be there before anything else. Witness Intel's 64-bit chip that won't run 32-bit code. AMD is stealing all the thunder with its 64-bit chip that is Windows32 compatible.

Far from the Clarion community not wanting C6, I, and the people I talked to, *certainly* wanted what it had to offer. It was just too complicated and difficult to convert my existing programs, on which I depend.

The new threading model scared me. I read the [white papers](#) (in particular, "Cooperative Threading In A Preemptive Environment")

and didn't understand the implications (or much else!), except that my previous method of coding appeared to be so crude and global-intensive as to be a complete "show stopper" for pre-emptive multitasking. So, I thought, forget Clarion 6 until next year...

Then the infamous C55 801-BIND error reared up endlessly, and various other errors kept me (and my clients) awake at night. I decided I would have to bite the bullet and convert. I printed the PDFs and pored over them – still all I could see was much extra code to protect my many globals, or a complete change in structure.

Then I found the *Global Cooperative Threading Extension*.

This extension allows me make C6 *work* like C55 and to convert my old apps to C6.1 *without* any problems with threading and without any immediate worry about critical sections etc. It makes C6 *behave* like C55. This is perfect for the initial conversion, and, later, when I need it (and have come to understand it), I can try out various sections with proper preemptive multitasking on known-to-work-previously code.

Note: The following discussion applies to single EXE applications in current releases of Clarion 6.1 If you try to use the Global Cooperative Threading Extension in a multi-DLL application, you'll get errors unless you're using at least Hotfix 9028, which as of this writing is still in beta.

Conversion went so very easily it was sad to remember all the earlier worrying and pain. I must emphasise, however, that before you begin you *must* back up your C55 apps and dictionaries.

I first downloaded all the new C6 third party libraries I use and installed them into C6 to give me the tools that I had in C55.

I then modified the threading template to default with cooperative threading on; I'll explain how to do that shortly.

In the global properties I put the Global "Cooperative Threading

Extension" (under Class ABC) and made sure that "Enable Global Cooperative Threading" checkbox was ticked. (Having modified the TPW file as I mentioned, above, when it added the Preemptive Procedure Extension, the "Enable Preemptive..." checkbox was unticked.)

I copied my app and dictionary into new C6 subdirectories and loaded them into C6. I accepted the conversion of the app, but *refused the conversion of the dictionary*. This is *very* important, because my imported app still pointed, using the full pathname, to the C55 dictionary and if I were to continue, the C55 dictionary would have been rendered useless to C55! (This is why you *must* make backups!) C6 then gave me the option to choose a new dictionary, so I specified the C6 dictionary which I *then* allowed to be converted. Other apps using the same dictionary don't have to convert the C6 dictionary again of course. But note that they *do* initially try to convert because they are still "looking" at the C55 version of the dictionary.

In the Global|File Control options I made the files "All Threaded".

The main app loaded and compiled without trouble except for my RTF edit window which was "deprecated". This has now been replaced by a Text Box with RTF ticked (RTF Text Control). I explain that technique below.

In my MDI-based apps I unlocked the Frame thread by ticking the "Enable preemptive threading on this Thread" in the Preemptive Procedure Extension. In an MDI application you must *not* lock the main thread.

The other "gotcha" was the change to the `Access:<FileID>.DeleteRecord()` call which has now acquired a parameter inside the brackets. If left out, `DeleteRecord` defaults to "Ask User" for each deletion. Since I do a lot of controlling of file size by deleting old records automatically, this was a no-no. So I had to go through and make all the instances into `Access:<FileID>.DeleteRecord(0)`. Tedious, but easy.

After that my application *worked*. C6 generated EXEs are *much* more stable than C55, and apparently faster too. And my new app *completely* handles the wheel-mouse. I have been abusing my program all day for three days, and it has never crashed! It appears that the underlying threading helps to make the program more stable even if I am working at the top level with a cooperative model.

But note that C6 is less tolerant of errors than C55, and more encapsulated because of the threading. For example, in C55, if you fetched a file record and you opened a child window on a new thread, you often didn't have to re-load the record (you *should* have, but you *could* get away with it!). C6 will not allow this because of the threading, and thus there is a new instance of the file in the other window.

I am sure there are many other minor issues, depending on what you have used, that can surface as a result of the real improvements C6 brings, but these have been the only ones I have come across so far on several fairly large and hand code-intensive apps.

I cannot recommend too highly converting to C6 for all the side benefits it brings, including robustness, speed, stability and the improvements to the IDE, especially in the dictionary editor. One cannot afford to be left behind with the technology because it becomes so difficult to catch up, but I was very fearful of the amount of work necessary to jump onto C6. The difficulty turned out to be a myth (in the short term). Of course, one has eventually to learn how to program with pre-emptive multitasking in mind, and anyway it is a better way to write programs. But it is *not necessary right now!*

I have been using Clarion for 17 years and love to see it getting so very good. I love the new IDE dictionary with everything visible and accessible quickly, and eagerly await the new 32bit IDE. Above all I love the stability that C6 brings to my programs.

The following two PDFs are available at SoftVelocity's web site, and this is where I learned about the cooperative template.

- [CooperativeThreadingInAPreemptiveEnvironment.pdf](#)
- [Preemptive&CooperativeThreadModels.pdf](#)

I mentioned earlier that I modified one template, and upgraded the old RTF control. Here are the details:

Implementation note: Modifying the Global Co-operative Threading Extension Template

When I first put the Global Co-operative Threading Extension into global, it *allowed* pre-emptive threading on *all* my 210 procedures(!), so I changed line 176 in Abthread.tpw (in Clarion6\Template) to default to %False. Now it reads:

```
#PROMPT('&Enable Preemptive Threading on this Thread', CHECK),  
  %EnablePreemptiveThreading,DEFAULT(%False),AT(4)
```

Then the Pre-emptive threading is not "on" in every procedure. (If I put in the Cooperative Template I expect preemptive threading to be the exception, not the rule.)

Implementation note: Upgrading the old RTF control

I deleted the old RTF box and control and populated the new RTF Text Control. In my case I use RTF just to show a special HELP since I need more than an ASCII viewer would give me, but I don't need the full edit and save etc.

Two things are different from the old RTF control – where I specify the file name to display, and how I let the user print with my own button. As usual, Clarion rose to the occasion. For the file name you need to declare a local CSTRING (I used `RtfFileName`), and then in the `LocalObjects|AbcObjects|WindowManager|Init` embed there is an embed point `RTF ?RTFFilenameInitialize`. Here's the code:

```
RtfFileName = 'MsgMerge.rtf'
```

Immediately after this I inserted into the embeds an `RtfAction` code template (from Class ABC) and chose the "Load" action and specified `RtfFileName` as the filename. It would be nice to specify the actual filename in `RtfAction`, but you cannot – it *has* to be a local (or global – that dirty word) `CSTRING` and be previously filled with the required file name!

For the printing, I again used the `RtfAction` code-template that has the print option. I populated a plain button and in the Accepted - Before generated code embed I did an insert and chose the `RtfAction` template again. Inside that, I chose an action (print) and a name (which doesn't matter) and lo! – it printed!

[Dermot Herron](#) grew up and went to school in Rhodesia, and earned an electrical-engineering degree at Capetown University. In the 1970s he worked for the Rhodesian Post Office, which was then also the telephone company. He was given the use of an HP9100, which was the very first programmable desktop computer, to help with the design of party-line telephones. He totally fell in love with computers then and there. Leaving the Post Office, he traveled for four years, worked in Canada (as a cabinet maker) and in England (as a microprocessor engineer) and then immigrated to New Zealand. But Dermot learned that once Africa gets into your blood you are doomed! Now he runs his own company in Johannesburg, writing software to send messages to mobile phones. He lives on a 19-acre plot with a river, and spends his off-time fixing things.

Reader Comments

[Add a comment](#)

Every once in a while one article pays for your entire...
This approach really works! I followed the simple...
Does this work for the Clarion Template Chain, or just ABC...
Hi David, I have used only ABC for years so know very...
In reply to my question on the C6 newsgroup, Stamos D...
I added the Global Cooperative Threading Extension on...
Multi-dll problems FIXED as of C6.1 build 9029. I asked...
Thanks Tom - I was going to ask... I hope all the...

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Reborn Free

CLARION
online

Product Review: The Ripley Code Commentor

by David Harms

Published 2004-09-10

The [Ripley Code Commentor](#), by Scott Ripley, is a nifty new utility that makes commenting code changes fast and easy, not just for Clarion but for other languages/development environments as well.

Let's say you've just been handed a whopper of a bug report, and you need to make a half dozen code changes to fix the problem. Probably the worst thing you can do is to just rewrite the code, obliterating the original source. Why? Sometimes bug fixes cause other bugs, and you may need to temporarily roll back your code, or at least understand how the original code worked. And even if you use version control software, simply seeing the changes won't necessarily tell you why they were made.

Ideally, you want to include comments with your changes, along with a copy of the original code (if you're deleting or changing something). And you should identify yourself by name or initials, particularly if you're not the only person working on that code base. But that's a lot of fairly tedious typing, and tedium is one of the main reasons code doesn't get documented the way it should. So what can you do? Ripley Code Commentor to the rescue!

RCC runs in the background and watches for global hot keys (which you can specify). You use the keys when you wish to add, change, or delete

source code. In the case of changing and deleting code, you highlight the code in question, press the hot key, and RCC cuts the highlighted code, modifies it with comments, and pastes the modified code back into the editor.

Installation

I obtained RCC by download from Scott's web site, www.ripleysoftware.com. (Actually at this time that domain forwards to <http://www.casinternet.net/~ripley/>.) The current release, as of this writing, is 1.18.6, and the download is about 1.2 MB.

I installed RCC without event. This product does not follow the Clarion 3rd Party Association [standards](#) for installation, but then it really isn't just a Clarion utility, so that isn't a problem, in my view.

Running RCC

When I launched the application, RCC displayed the window shown in Figure 1.

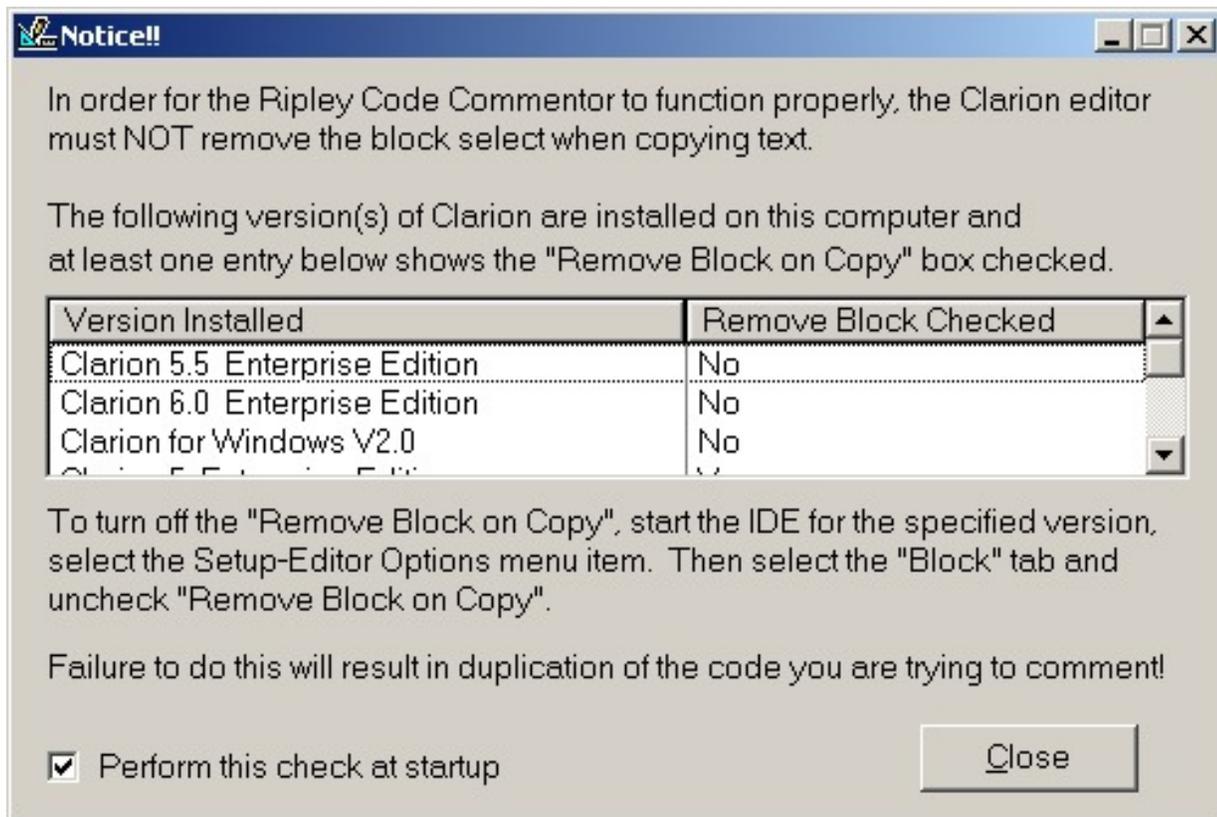


Figure 1. Clarion configuration checking.

This step is important because of a peculiarity of the Clarion source editor, which has a "Remove Block On Copy" option. See Setup|Editor Options|Block, for the dialog shown in Figure 2.

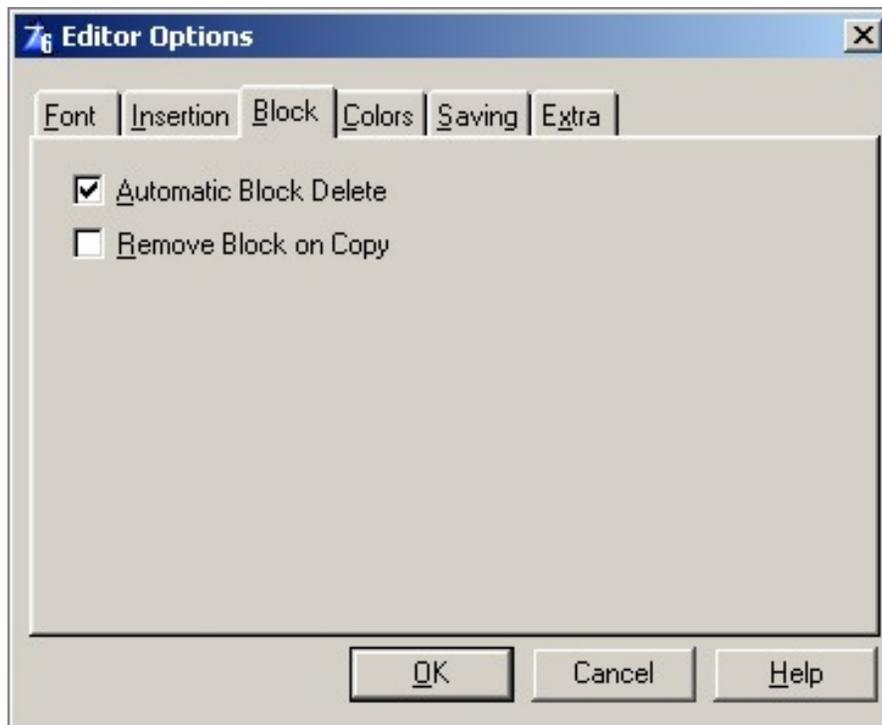


Figure 2. Correct settings for Remove Block On Copy

Failure to use the correct setting will result in duplicated source code. My advice is to set this and forget it. I never could understand why I would want to delete the text I'm copying anyway. If you're using RCC in an environment other than Clarion you can disable this check; if you later need it, just turn it back on from the configuration dialog.

Once you've got past the Clarion check and a splash screen, you'll see the main application window, as shown in Figure 3.



Figure 3. The Ripley Code Commentor

Using the default configuration, clicking on any other application or the Windows desktop will cause RCC to minimize to the system tray. To restore the RCC main window, just right click on the icon and choose Restore, or double click on the icon.

RCC provides five primary functions, as noted by the hot keys and their associated buttons on the RCC main window.

The Add function

To use the Add function you place the cursor at the desired insertion point and press the Insert hot key (the default is F6). The Add function was designed for pasting in code – you can use this function without copying some code to the clipboard first, but if there is something in the clipboard it will get pasted. The rest of what RCC pastes depends entirely on what you've specified in the Initials, Task Num/ID, and Task

Desc fields.

```
! Added DH 09/08/2004 #1 Change in overseas delivery option
! End Add DH 09/08/2004 #1 Change in overseas delivery option
```

The Change function

To use the Change function you select the code you want to change, and press the hot key (default is F7). RCC makes a copy of the code, commented out, so the result looks something like this:

```
! Changed DH 09/08/2004 #1 Change in overseas delivery option
! BRW1::View:Browse{prop:sqlFilter} = Filter
! Brw1.ResetFromBuffer()
! to
BRW1::View:Browse{prop:sqlFilter} = Filter
Brw1.ResetFromBuffer()
! End Change DH 09/08/2004 #1 Change in overseas delivery option
```

The Delete function

The Delete function simply comments out the selected code and brackets it with delete comments:

```
! Deleted DH 09/08/2004 #1 Change in overseas delivery option
! BRW1::View:Browse{prop:sqlFilter} = Filter
! Brw1.ResetFromBuffer()
! End Delete DH 09/08/2004 #1 Change in overseas delivery option
```

The Last Copy function

The Last Copy function is a lifesaver, which probably wouldn't be necessary if the Clarion editor had a real multi-level undo capability. But the editor doesn't, which means that pressing Ctrl-Z after you've used one of the other functions will not restore your original code. Press the Last Copy hot key (default F9) and RCC pops up its main window, and displays the original source, which it has just placed back in the clipboard. If you want to paste this text you can, or you can select a portion of the original text, and do a copy and paste, or you

can just click anywhere but the RCC window to send RCC back to the background.

The (select) Language function

Languages supported include Clarion, the Clarion template language, Java, C/C++, HTML, VBScript, ASP, Oracle, dBase, DOS batch files, and Modula-2. You can easily add support for other languages by entering the beginning and ending comment characters. For instance, the Clarion language uses a ! beginning character and no ending character, and the Clarion template language uses #!. HTML uses <!-- and --> to denote comments.

Configuring RCC

To configure RCC to your particular requirements, choose Setup/Configure from the File menu. This brings up a configuration window where you can change hot key assignments, and customize language support (i.e. specify which language is current, and which is the startup default). You can also get quite detailed in the setup of user comments, using a system of tokens and templates. There is a Miscellaneous tab for things like increasing the delay between the copy and paste (if too short, this may cause problems with some machines), minimizing behavior (you can set RCC to stick around instead of minimizing when you switch away), and so forth. Finally there is a tab just for some setup documentation regarding the Clarion block select setting.

Visually, RCC is nothing fancy, and it doesn't need to be since this is a programmer's utility. The user interface could use a little tightening up, however, as there are some inconsistencies. For instance, double-clicking brings up a change form on most list boxes, but not on the languages list box. And right-click context menus seem to be missing completely.

Version 1.18a introduced the ability to automatically check for updates

using CapeSoft's NetTalk package. RCC uses NetTalk's Async mode so a missing or bad Internet connection will not cause the application to hang. The check interval is every two hours. As of version 1.18.3 you must explicitly turn on this option via the Miscellaneous tab of the setup window. Apparently some users were concerned about the application unexpectedly making use of the Internet. You can see in Figure 3 that while I'm using version 1.18.2, a new release, 1.18.5, is available. Clicking on the link brings up the RCC web site.

Curious about RCC's multi-language capability, I tried it with Eclipse, which I use for Java development. Aside from a minor tabbing issue (pasted source was tabbed further right than I wanted, not a big deal since Eclipse has a source formatter), RCC worked just fine.

Documentation

RCC comes with minimal documentation, in the form of an RTF file which is called from the RCC Help menu. While RCC is a fairly straightforward utility, and the average developer can probably use it successfully out of the box, not all functions are covered in the text. For instance, there is no mention in the documentation of the template system and how user-defined fields work, and there are no screen shots at all.

Technical Support

I emailed Scott several times during regular office hours, once to report what I thought was a bug but which neither Scott nor I could reproduce, and several times with questions about the product. In all cases Scott's response was nearly instantaneous.

Summary

In the first draft of this review, which I sent to Scott for his comment, I mentioned that I'd like to see multi-line comments (right now all field

data is put on a single line), and perhaps an option to have a popup window appear when the hotkey is pressed so I could choose from recently-used task numbers/descriptions. Just after midnight Scott sent an email notifying me that "Version 1.18.6 implements multi-line comments and a hot-key for recently used task ID/descriptions." So there!

In the second draft I added my comments about user interface inconsistencies, so by the time you read this those will most likely be fixed also. I really like the detailed revision history available on the Ripley Software web site. This is one actively maintained and improved product.

Overall, the Ripley Code Commentator is an excellent little utility that should make code documentation, particularly in multi-programmer shops, a whole lot easier.

Available from www.ripleysoftware.com for US\$20 per license, through ClarionShop, eSellerate, and via PayPal from the author. The unregistered version has a two week trial period.

Scott Ripley
RR 3 Box 269A
Washington WV 26181

Rating

Ability to do the task	
Easy of use	
Ease of installation	
Documentation	
Technical support	
Black box DLLs/LIBs	n/a
Overall Rating	

[David Harms](#) is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of Developing Clarion for Windows Applications, published by SAMS (1995). His most recent book is [JSP, Servlets, and MySQL](#), published by HungryMinds Inc. (2001).

Reader Comments

[Add a comment](#)

**In reference to your comment, "I never could understand...
In the programs that behave that way, I simply copy twice..."**

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)

online sales and delivery
for your applications & tools

Developer **PLUS**

Break Dancing With Clarion

by Geoff Bomford

Published 2004-09-16

I'm happy to say that Clarion 6.1 has introduced some improvements to the report designer! These aren't the sort of changes that will, on their own, make you want to run out and upgrade to C6.1, but they are important advances nevertheless. In the first article in this review I'll cover the use of pre-Clarion 6.1 techniques for performing some common report break tasks, and in the next article I will discuss the use of the BreakMgr Class, a new tool in Clarion 6.1 for handling the same tasks a little better!

What is a Report Break?

In reporting it is common to have records grouped together by a break variable, and the group of records will commonly have a break header and break footer. For example a Product Sales Report might list records from the InvoiceDetails file, grouped by Product ID, the break variable. The break header might contain the ProductID and Product Name, and the break footer might contain totals for number of times the product has been sold and the total sales for each product.

Three of the most frequent questions asked about reporting in Clarion are:

1. Where can I put the code that executes before the break header/footer?
2. How can I perform calculations on totals, before they print in the break footer?
3. How can I reprint a break header at the top of a new page?

I'll try to answer these questions with standard Clarion reporting techniques.

Formatting break headers

With standard Clarion reporting all three of these events happen with the `PRINT()` statement. But, the `PRINT()` statement can *only* be used with a report detail, for example `PRINT(Rpt:Detail)`. There is no option for `PRINT(Rpt:ProductBreakHeader)`.

The Clarion report engine takes *any* `PRINT()` statement and performs the actions shown in Table 1 *without* the option for programmer input.

Table 1

For the current record Did the break variable change?	
No	Yes
Apply detail formatting	Apply footer formatting
Print detail	Print previous footer(s)
Update totals	Clear totals
Page overflow	Apply header formatting
	Print new header(s)
	Apply detail formatting
	Print detail
	Update totals
	Page overflow

Since the report engine handles all these tasks without programmer input, any code you want to execute before the footers and headers are printed has to be placed *before* the `PRINT()` statement.

The application accompanying this article was created with Clarion 5 and contains a series of report procedures designed to illustrate this printing process and answer the questions posed above.

Report1 procedure

Report1 is a procedure showing a simple ABC report based on the Product and Detail files from the Invoice example application. `Pro:ProductNumber` is the break variable used for the break header and footer. It is simply the starting point for the following examples.

Report2 procedure

Report2 builds on Report1 by demonstrating how to format a string in the break header. In this case, I want to change the color of the `Pro:Description` header to blue for any products where the description begins with the letter "B". The code is quite simple, and goes in the Source, Report2 TakeRecord code embed, priority 5001:

```
SETTARGET(Report)
IF SUB(PRO:Description,1,1) = 'B'
  ?PRO:Description{Prop:FONT,3} = Color:Blue
ELSE
  ?PRO:Description{Prop:FONT,3} = Color:BLACK
END
SETTARGET
```

Note the `SETTARGET(Report)` statement. This tells Clarion that any properties I set after this statement will affect controls on the Report. This is *essential* in reports, because the default TARGET is the procedure's progress window! Following that statement I can set the properties of any controls on the report, in this case the font color. Don't forget to issue `SETTARGET` (with no parameters) following this code, otherwise you might get some unexpected effects for other controls when generated code attempts to update controls on the report progress window..

The above code is exactly the same for ABC and Clarion template based reports, and the embed point is the same – *before* the `PRINT()` statement! For ABC this means anywhere in the TakeRecord embed point with a priority of less than 8000.

Performing Calculations on Break Footer Totals

Performing calculations on totals in break footers is a little trickier. The code has to be placed *after* a `PRINT()` statement, since it is the `PRINT()` statement that updates the totals! Can you see a *Catch 22* here? The `PRINT()` statement updates footer totals and then prints the footers, and there is nothing you can do about it, right? Well actually there is a trick to doing this, which I'll now explain.

Report total controls have a parameter called `USING`, which is a variable used by the report engine to store break totals. It is only active on controls which have a `Total` type specified, as shown in Figure 1.

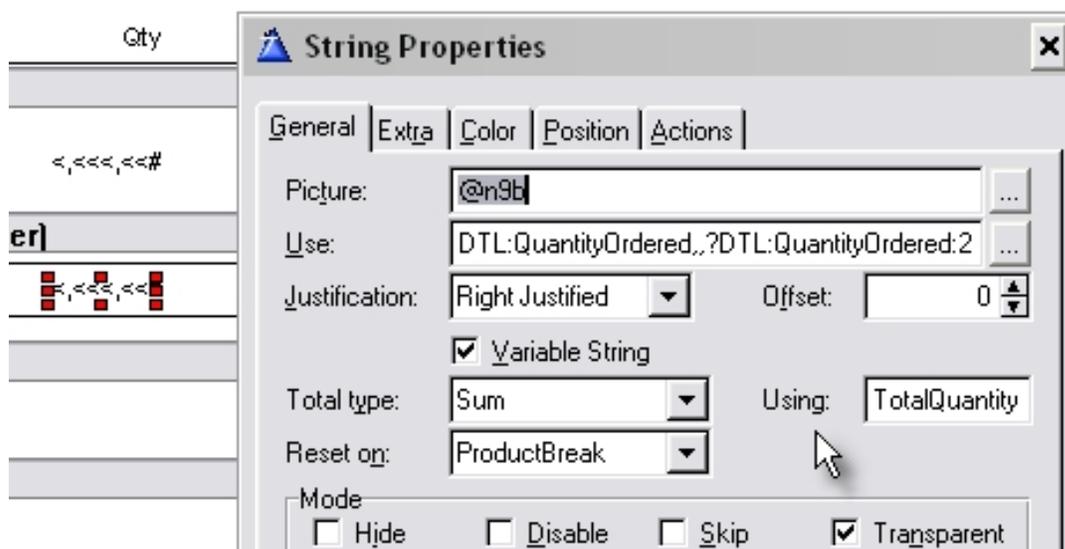


Figure 1. Report3 – Specify a "Using" Variable

In this case I have created a break total for `Dtl:QuantityOrdered`, and I want to use that total within another calculation. So, I created a local variable called `TotalQuantity` and specified that as the `USING` variable for this total. I've done the same for `ExtendedPrice`, and I now want to calculate a weighted selling price for each product. The calculation is:

Weighted Selling Price = Total Extended Price / Total Quantity Ordered

Or...

Local:WeightedAveragePrice = TotalExtended / TotalQuantity

Report3 procedure

Report3 demonstrates the wrong way of implementing calculations on totals with a USING variable. You can see that I have placed the code to perform my calculation in an embed after the code generated by a "Formula Field", and before the PRINT() statement. What could be wrong with that?

Well, from Table 1 you can see that totals are updated *after* the PRINT() statement, so the total calculations are going to be all out of synchronization, and if you run Report3, you will see that they are!

```
! Where you would expect the Total Calculations to be updated
Local:WeightedAveragePrice =
TotalExtended / TotalQuantity
```

Should I then move the calculation to *after* the PRINT() statement? Brrrrrp! Incorrect answer! If you look at Table 1 again, you will see that Clarion's break handling will print the previous break footer, then the new break header and detail, and only *then* are the totals updated!

In Report3 I have added my Local:WeightedAveragePrice control to the break header, detail and break footer, so I can try to understand what is going wrong. Experiment yourself by moving the location for this calculation to after the PRINT() statement and you'll see that it just doesn't work!

Report4 procedure

There has to be a way to do this doesn't there? Well, yes, there are three ways. You can hand code all your totals and keep track of break variables yourself. That works, but is hardly RAD! Or, you can use Clarion 6.1, which has new technology for this, which I'll cover in the next article. Or, you can think like the Clarion report engine, or whoever designed it!

Report4 demonstrates the last of these techniques. If the PRINT() statement updates the totals, what I need is *another* PRINT() statement, *just to update the totals!* In Report4 I have added an empty detail called TotalDetail, given it a USE variable ?TotalDetail, and filtered it from the report properties so it is never printed by the templates. (Refer to [Steve Parker's](#) excellent articles on reporting to see how this is done if you are not sure). The height of the

TotalDetail has been left at Default and the detail contains no controls.

The real trick to this is knowing that when you `PRINT(Rpt:TotalDetail)`, or any empty detail with default height, the report engine *does nothing*, except – update totals! Headers and footers don't get printed; there are no page headers, nothing. If you create a report with an empty detail, default height and run it, your progress bar will indicate that the report has processed the records, but when you get to the end there will be nothing to preview and the procedure will just close.

So, how do I use this bit of knowledge? First I have to make a change to the total variables, to ensure that the totalling is now performed on the TotalDetail, not Detail.

Every `PRINT()` statement updates the Totals; I only want them updated when I `PRINT(TotalDetail)`. So I go to the properties for every total and make sure that *only* TotalDetail is used for the "Tallies". See Figure 2.

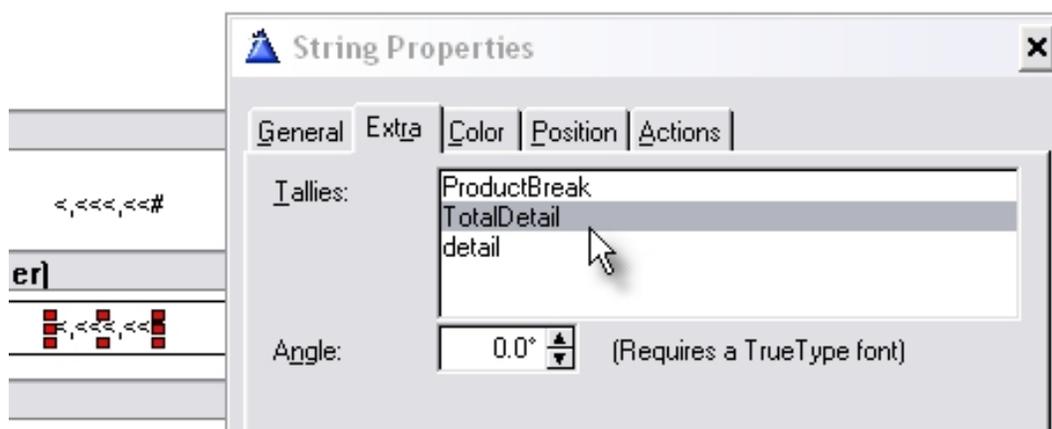


Figure 2. Specifying TotalDetail

Now, place the following source code in the TakeRecord embed before `PRINT(Rpt:Detail)` (say, priority 5001):

```
! Force USING variables to be updated, by printing the TotalDetail
PRINT(Rpt:TotalDetail)
Local:WeightedAveragePrice = TotalExtended / TotalQuantity
```

I will manually `PRINT(Rpt:TotalDetail)`. This updates the USING variables and I can access the correct values in my calculations. The report now has the correct result in the footer!

This also allows me to do other cool things with break footers, based on break totals. Consider this code, in the TakeRecord code embed, priority 5001:

Source, Report4 - TakeRecord Code, Priority 5001

```
! Use the USING variables to conditionally format the Break Footer
SETTARGET(Report)
IF TotalExtended > 100
  ?Box1{Prop:Fill} = Color:Yellow
ELSE
  ?Box1{Prop:Fill} = Color:None
END
SETTARGET
```

Using similar code to that shown above in Report2, I can now format the footer to highlight information based on break totals. In this example, I am coloring a box in the *footer* yellow when total sales for a product exceed \$100. Easy, eh?

How can I reprint break headers at the Top of the Page?

This question is usually answered by "You can't!" The page overflow referred to in Table 2 is another automatic task handled by the report engine. Whenever the report engine writes to the report the page overflow process performs the tasks listed in Table 2.

Table 2

For the Current Report Output			
Is there enough room at the bottom of the page?			
No		Yes	
Does the Previous Print Output have a " <i>With Next</i> " value that includes this output?		Does this Print Output have a " <i>With Next</i> " value that includes this output?	
No	Yes	No	Yes

Print a New Page	Print New Page	Print this output!	Save this output for the <i>next</i> PRINT() statement
Print Page Headers	Print Page Headers		
Save this output for the <i>next</i> PRINT() statement	Print <i>previously saved</i> output		
	Save this output for the <i>next</i> PRINT() statement		

Clarion's page overflow process only determines when a page break is needed; it does not automatically re-print break headers, only page headers.

"With Prior" and "With Next" are settings used by Clarion to avoid widows and orphans on a report. A widow is a header at the bottom of a page with no detail records following, and an orphan is a detail, or footer, at the top of a page without related break headers. If you look at the preview of Report1 you will see an orphan on Page 3.

Product Sales Report

Total:	1	5
--------	---	---

Figure 3. Report1 showing an orphan on top of page 3

Figure 3 shows a Product group footer total printed at the top of the page, but there is no information to suggest which product it refers to.

Report5 procedure

Report5 demonstrates the use of WithPrior and WithNext to avoid widows and

orphans. It is severely limited however because there is no option to reprint break headers at the top of a new page. Report5 (Figure 4) shows a consequence of this on Page 2.

Product Sales Report			
	0000002	20	\$2.50
Total:	7	344	

Figure 4. Report5 showing missing header, top of page 2

Although this is not technically an orphan, it still looks unprofessional to have records printed at the top of the page without the appropriate headers.

WithPrior and WithNext are specified in the properties of any report band printed within the Detail area. It is sometimes necessary to juggle with the values of WithPrior and WithNext to get the desired results, and sometimes the results can be unexpected, so test carefully. In this example I have specified a WithPrior value of 1 for the break footer. This means that when the break footer prints it will have at least one detail record with it. If I specify a value of 3 for this WithPrior, then three detail records will print at the top of page 2, prior to the break footer printing. Try it and see for yourself!

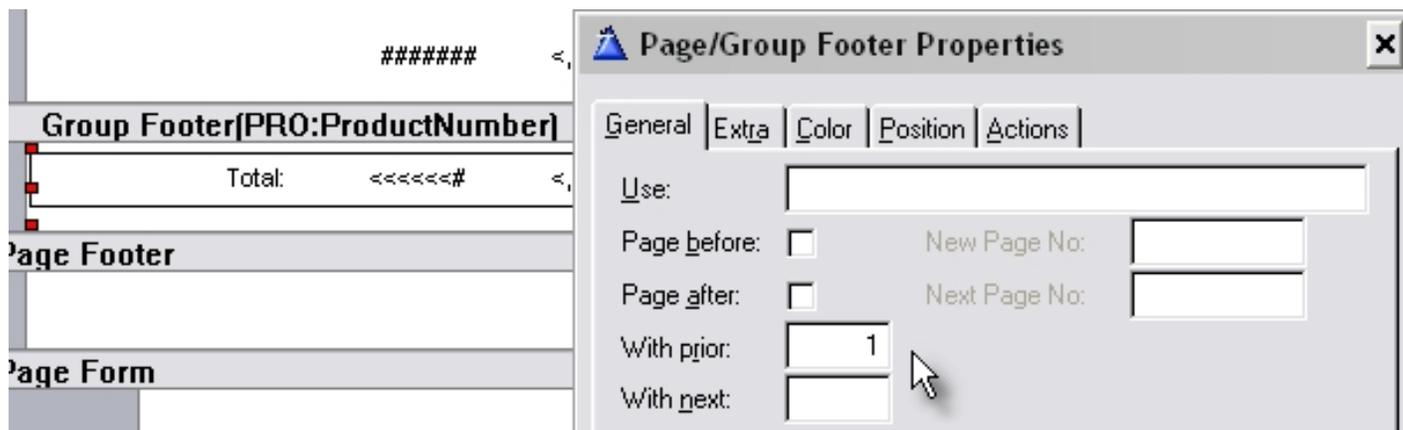


Figure 5. Specifying a "With Prior" for the group footer

Clarion's page overflow management is especially frustrating because it is all handled automatically, and there is no way for a programmer to determine

when a new page has been printed. So, there is no way for a programmer to manually re-print the break headers.

There is at least one third party product that tries to provide this functionality. I believe similar functionality should be built in to the Clarion report engine/template! Does the `BreakManagerClass` from Clarion 6.1 provide the solution? The answer to that will be in my next article, which takes a look at how the `BreakManagerClass` handles the same three tasks discussed in this article.

[Download the source](#)

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

online sales and delivery
for your applications & tools

Developer **PLUS**

Dynamic Table-Based Tagging

by **Chris Laurie**

Published 2004-09-17

For a while now I have been using the excellent tagging class from the [article by Steven Parker](#). It has the sort of "point and do" simplicity that I like as a lazy programmer. But recently my users' requirements for tagging became more sophisticated. The tagging class has a two drawbacks in terms of my new requirements: saving the tags between sessions is difficult, and you have to add a global instance of the class for each set of tags, forcing a complete regenerate and compile of the entire app each time you want to add more tagging. Not fun in a 200 procedure app on a slow machine...

With the advent of C6 I was also worried about the fact that the tags are stored in global queues and whether this method is thread safe. The thought of hacking the class with a synchronization object left me cold.

So I decided to bite the bullet and develop a new tagging method with the following features:

- Make the tags persistent between sessions of my application;
- Link tagging to a user so that his tags follow him around (a "gaggle" of tags?);
- Allow for "global" tags (seen and used by all users);
- Create tag sets dynamically (without having to create a new table or global variable);

The obvious route was to store the tagging information in a table. I only want to use a single table to store all tags, for all users and uses. The `UserTags` table has the following layout:

```

UserTags      FILE, DRIVER( 'TOPSPEED' ), PRE(UTG), BINDABLE, CREATE, THREAD
KeyUserTags   KEY(UTG:User, UTG:TagArea, UTG:TagValue), NOCASE, OPT
Record        RECORD, PRE( )
User          CSTRING( 20 )
TagArea       CSTRING( 40 )
TagValue      CSTRING( 100 )
              END
              END

```

The given sizes are not important, you must just make sure you have enough space in each of the columns to store your information. The columns do the following in the tagging system:

- **User** – Stores the name of the user for which I am recording the tag. This assumes that I have a method of tracking users of the program. I can also specify a user called 'GLOBAL' or even make up different sets of global tags as I go along.
- **TagArea** - This column is used to identify what I am tagging. I make this up as the need arises. The only caveat is that I must make sure that I always use the correct value. For example, when I'm saving tags for members of an organization, this column is primed with 'MEMBER'
- **TagValue** – This column contains the unique identifier for the record that I want to tag, like the member record's unique SysId. The column size allows for tagging of long strings as well, such as names.
- The **KeyUserTags** key makes looking up and manipulating tags quite efficient.

My example tagging app uses the data from Club Manager example that comes with Clarion 5.5. I create global values for my most commonly used tag areas; in the case of the Club Manager app, they are as follows:

```

EQU:TagAreas  GROUP, PRE(EQTA)
Members       CSTRING( 'MEMBERS<0>{32}' )
MemberTypes   CSTRING( 'MEMBERTYPES<0>{28}' )
Committees    CSTRING( 'COMMITTEES<0>{29}' )
States        CSTRING( 'STATES<0>{33}' )
              END

```

This "breaks" my rule of not wanting to define any global variables, although these are really equates, as their values don't change. (I have not found out how to declare equates in the dictionary so that they are available from the populate column box in the embeditor). I actually do not need to declare these equates. I only do it because I make lots of typo errors and this way the compiler will help me check that I am consistent. If I use an equate I cannot

have the situation where my code creates tags for 'MEMEBERS' and tests for 'MEMBERS'.

Manipulating the tags

I also need a few functions and procedures to manipulate the tags and make them easy to use. These procedures have all been declared globally. None of these is complex – they just do simple lookups and updates on the `UserTags` table.

- `TagAnyTagged (PassedUser, PassedTagArea)` – Returns 1 (True) if there are any tags for this user for the specified tag area or otherwise 0.
- `TagClearAll (PassedUser, PassedTagArea)` – Deletes all records in the `UserTags` table for a specific user and tag area.
- `TagClearOne (PassedUser, PassedTagArea, PassedTagValue)` – Deletes a specific tag.
- `TagCount (PassedUser, PassedTagArea)` – Returns the number of tags for a user and tag area.
- `TagIsTagged (PassedUser, PassedTagArea, PassedTagValue)` – Returns 1 (True) if a specific record is tagged.
- `TagSetOne (PassedUser, PassedTagArea, PassedTagValue)` – Creates a specific tag record.
- `TagToggle (PassedUser, PassedTagArea, PassedTagValue)` – Will delete a tag if it exists or create one if it doesn't.

Putting it together

To make the example app run, copy the APP and DCT files into your C5.5 ClubMgr directory (or make a copy of that directory elsewhere), load the application, compile and run. I have used the tags in a number of different ways to demonstrate the flexibility of this approach.

A small aside on using the example app: The demo data for the members table in club manager example has almost all the members in the state of 'FL'. I randomised this in order to make the data more realistic and demonstrate the application better.

WhoAmI procedure

This procedure would be replaced by your own logon system. It is crucial to the scheme that we identify each user as they use the system. This is what allows tags to follow the user around.

BrowseMembers procedure

This procedure demonstrates several different uses of the tagging system. The following bits of code are interesting:

- **SetBrowseFilter routine** – This routine constructs the browse filter and calls `Brw1.SetFilter` with the constructed filter. Because I am calling a procedure in the filter `'TagIsTagged(GLO:User, EQTA:Members, MBR:MemberNumber)'` I must make sure that all the variables and procedures are bound.
- **ThisWindow.Init** – This code contains the `BIND` statements necessary to make the filter work.
- **ThisWindow.Reset;**
 - `?Loc:ShowTaggedStates {PROP:Disable} = CHOOSE(TagAnyTagged(GLO:User, EQTA:States), FALSE, TRUE)` – This sets the disable status of the check box for the `Loc:ShowTaggedStates` variable. This check box is only enabled if the user has any states tagged (in the `BrowseStates` screen). The checkbox variable in turn is used to add a specific component to the filter in the `SetBrowseFilter` routine (see above).
 - `?Loc:ShowMemberTypes {PROP:Disable} = CHOOSE(TagAnyTagged('', EQTA:MemberTypes), FALSE, TRUE)` – This shows how I have used the tagging system to create "global" tags that are shared by all users. Since the user is empty, the tag will be visible in all instances of my program, by all users.
- **TakeAccepted on the ?ShowTaggedMemberTypes button.** This code shows how to retrieve a list of the current tags of a user for a tag area.
- **EVENT:AlertKey in TakeFieldEvent on Browse1.** This code allows the user to tag a record in the browse and the scroll down to the next record. I had some problems getting this to work as it interfered with the locators. See the code in `BRW1::Sort0:Locator.TakeKey`, which intercepts the space key. Unfortunately I have to add this bit of intercept trickery to all the locators. There probably is a better way to do this.

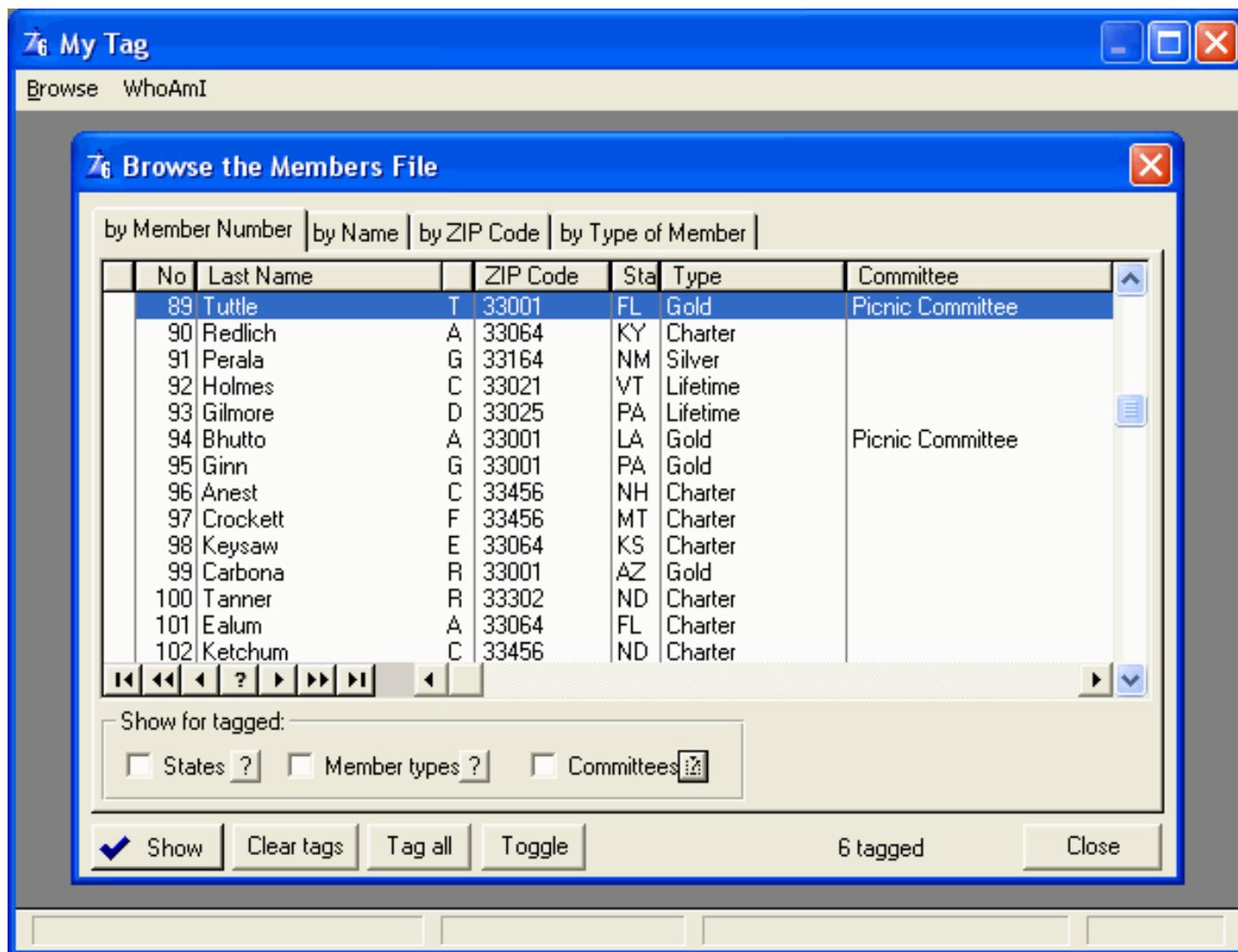


Figure1. The basic members browse.

Figure 1 shows the basic browse members screen. There are six records tagged but they are not visible in this page list.

Complex filtering

The filtering in `BrowseMembers` allows you to limit the list of members based on whether a member has a specific attribute that was tagged in another browse. In other words, it is easy to only show members in the list that serve on any of the tagged committees. By combining these filters the user can construct a complex set of criteria by tagging various attributes in the committee members, member types and states browses and then see only the members that satisfy all the criteria (or for that matter any of the criteria). This is accomplished by the check boxes in the show tagged group. Similarly, a print procedure could loop through the member table and select rows for printing based on which

attributes have been tagged. Or you could create a set of tags that would be read by a printing routine.

Dynamic Global Tags

The method of manipulating the user to make a set of tags global gives the system added power and flexibility. You could store tags based on the user's security settings by changing the `UTG:User` field to 'SUPER' (all users with this setting will then see and use the tags) or separate global tags for different shifts 'MONDAY' or 'NIGHTSHIFT'. The permutations are endless.

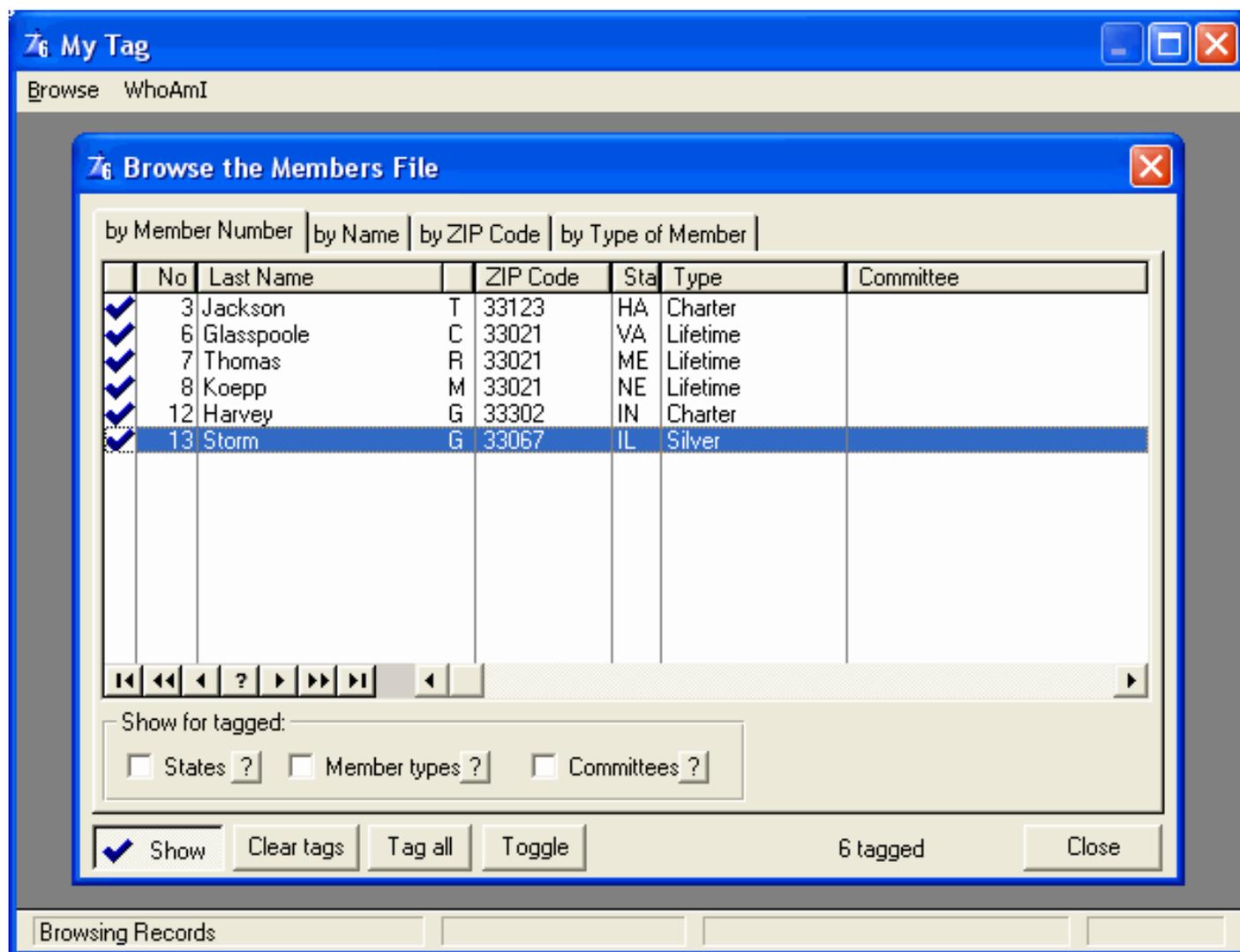


Figure 2. The browse with the filter applied under the Show button.

BrowseStates procedure

This procedure shows how easy it is to use the tagging system to manage pick lists. The other browse screens uses a checkbox (made into a button) to switch on a filter that will limit the list to only show the tagged records. The `BrowseStates` procedure uses the popular alternative of showing two lists: one list of available choices and another of actual items chosen. This is very easy to do with the tagging system.

The available states list on the left is a browse filtered for states, which have not been tagged. Filtering is done in the `ValidateRecord` method. If the current user tagged a state, it is filtered.

The actual states chosen list on the right is a browse of the `UserTags` table that has been range limited for the current user and then filtered on the `EQTA:States` tag area.

There is a bit of magic in the browse on the right: the custom join between the `UserTag` and the `States` tables. Since the tag table only contains the state code and not the name, I use this custom join to pull the name of the state into the browse. You can use the same method to bring in the name of a member when you are storing the `SysId` of the member row.

The pick button (`>>` at the top between the lists) creates a tag for the highlighted state on the left and reset the screen. This causes the browse lists to apply its filtering and voilá: the state appears only in the right hand list.

The remove button (`<<` below the pick button) deletes the corresponding row from the `UserTags` table that is highlighted on the right, and then resets the screen. It would be simple to achieve this in a drag and drop scenario.

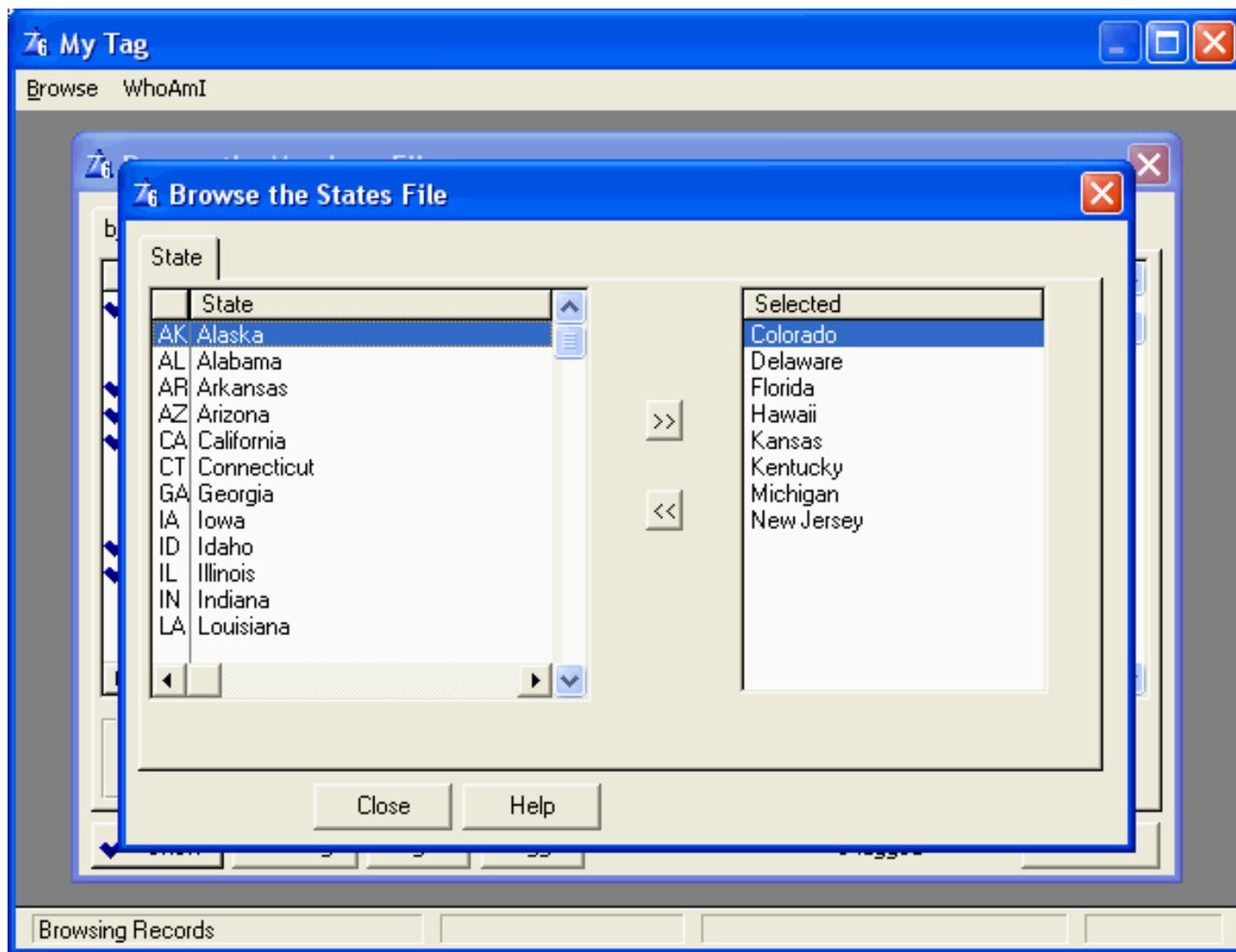


Figure 3. Implementing the tag list

Figure 3 shows how the tag list was implemented. The list on the right is a browse on the actual `UserTags` table. The list on the left contains a filter in the `ValidateRecord` method. See my comments below about using a custom join to pass this filtering to the view.

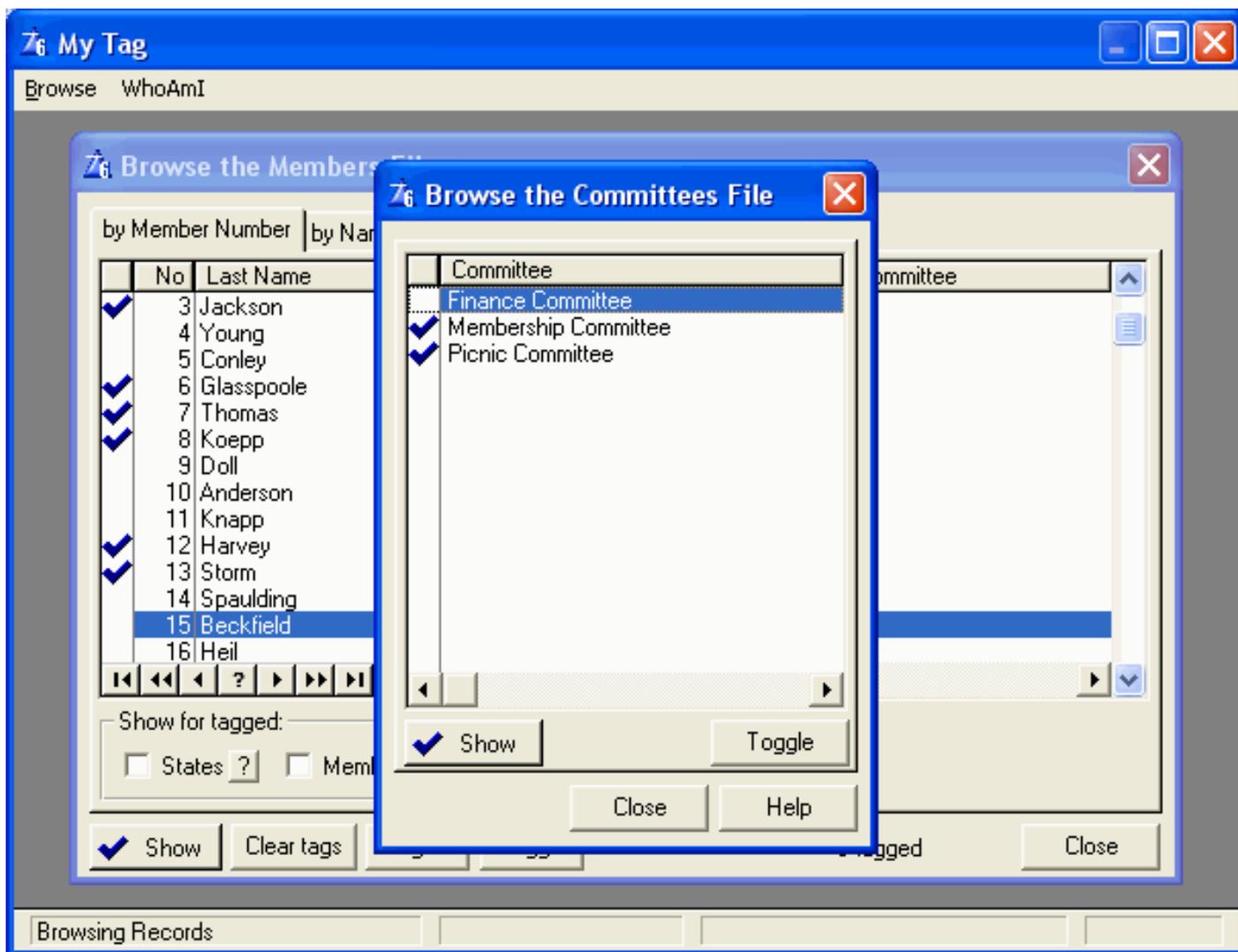


Figure 4. A basic tagging screen

Figure 4 shows a basic tagging screen, where the user can tag a selection of committees. The tags by themselves have no meaning. It is when you filter the member browse by the tagged committees that it really begins to work.

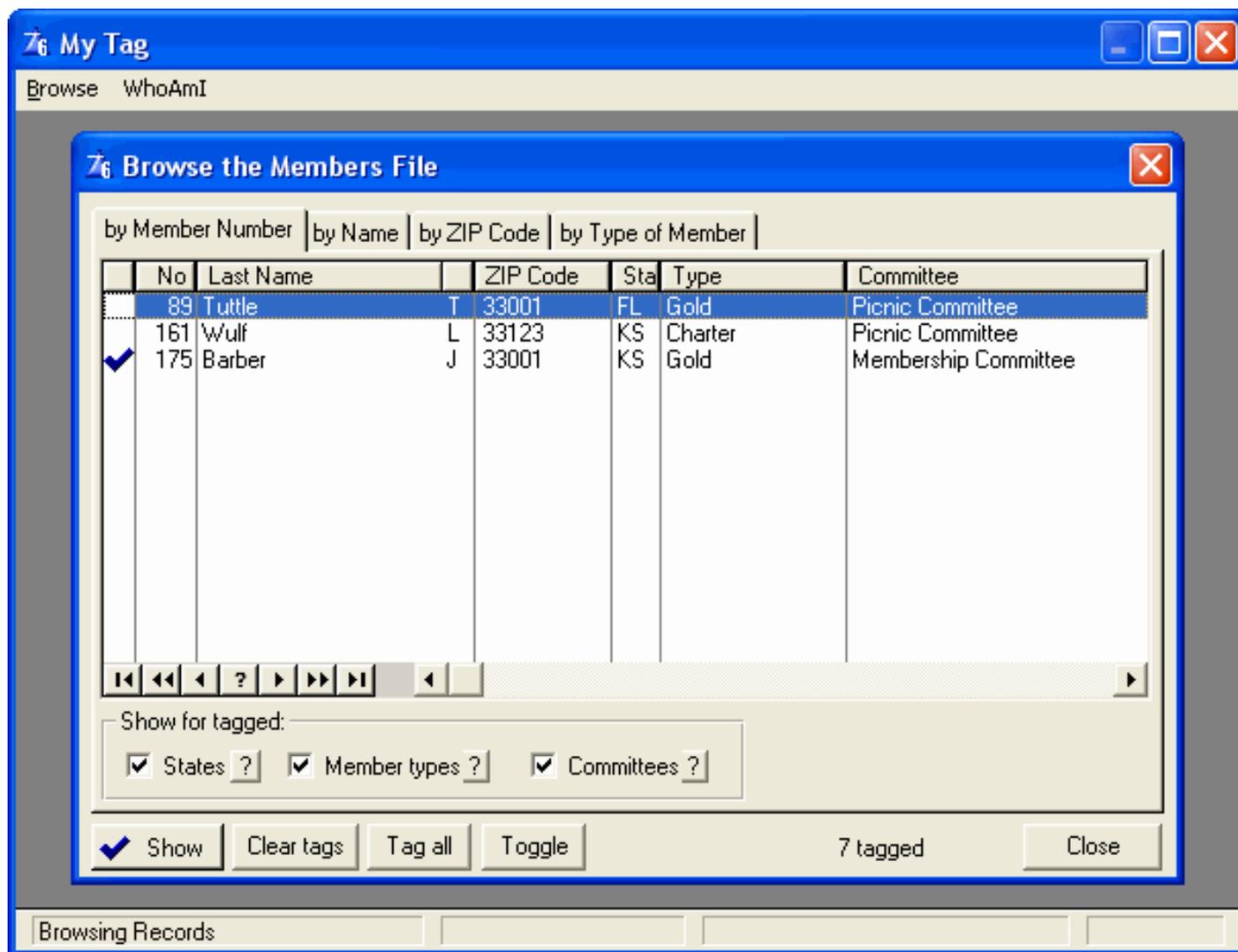


Figure 5. The tagging system in action

Figure 5 shows the effect of the tagging system. Notice that all the Show for tagged checkboxes are on (States, Member types & committees) and that the list of members are now reduced to those that satisfy all the tagged criteria in terms of tagged states, and member types and committees. Also note that only member 'Baber' has been locally tagged in this subset, even though there are seven members tagged in total. If you press the Show button then the list will be reduced to just the one entry.

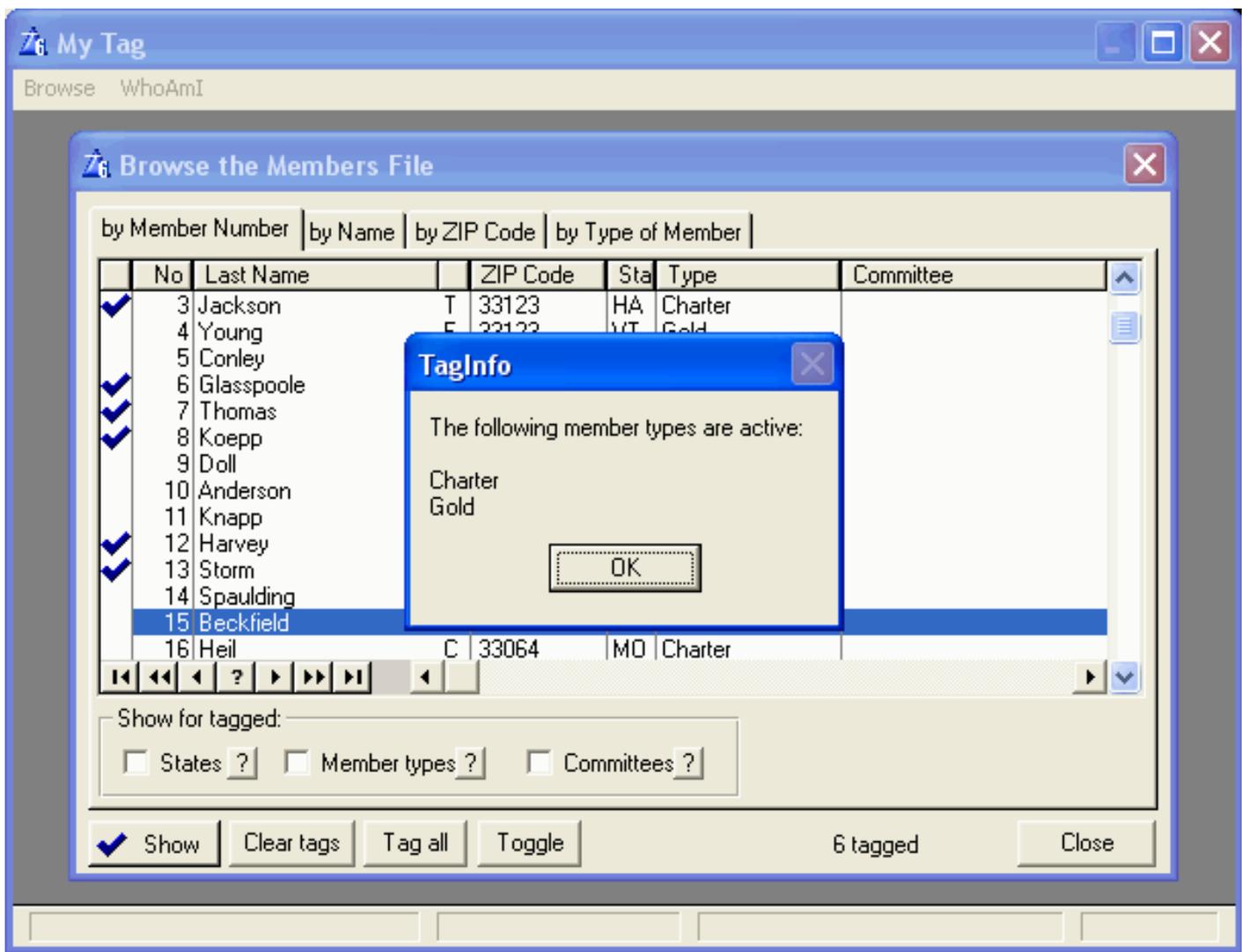


Figure 6. Reading global tags

Figure 6 shows how read the global tags for member types to make up a message.

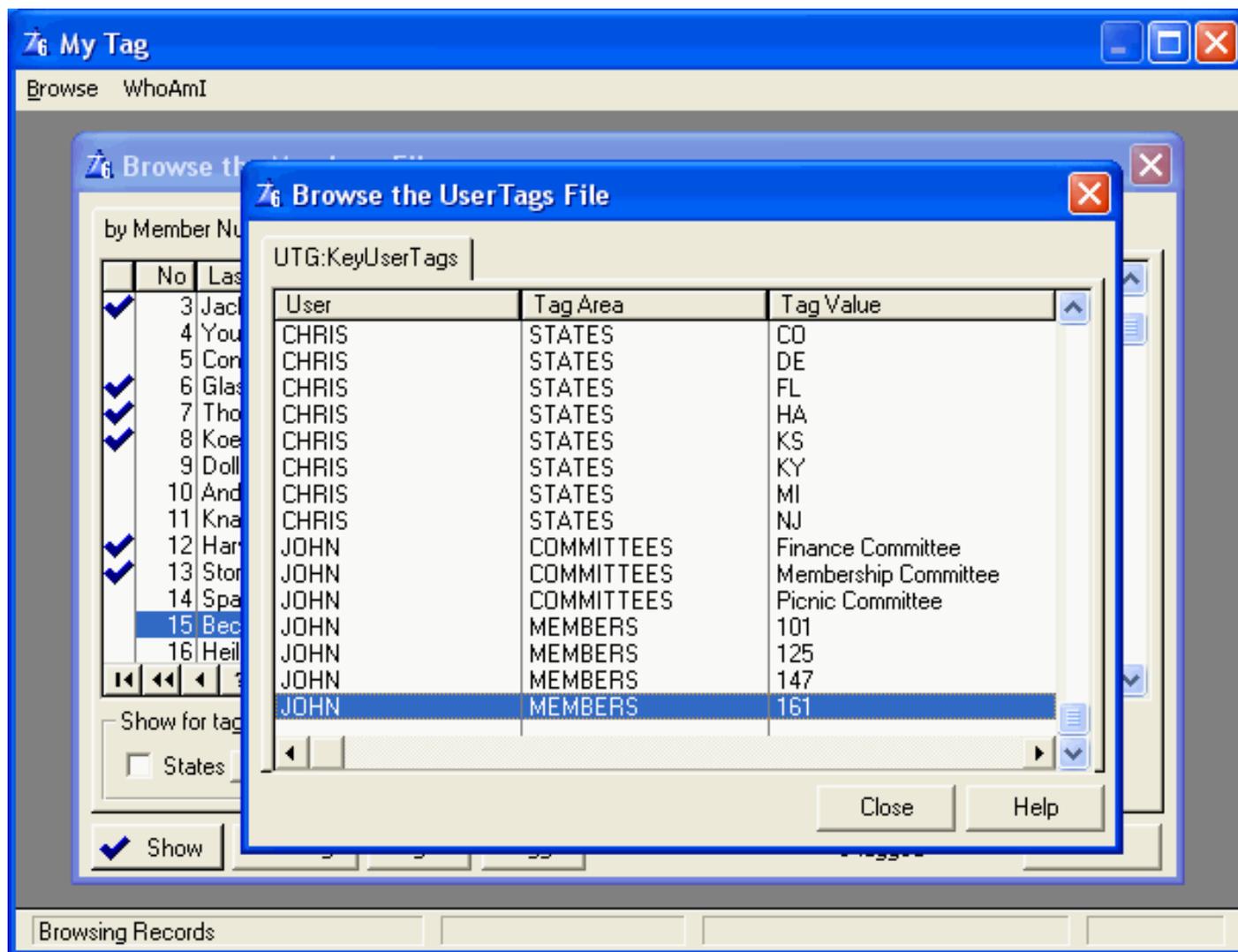


Figure 7. The UserTags table

Figure 7 shows the contents of the UserTags table. Because of Clarion's automatic type conversion, I can store any arbitrary value in the TagValue column.

It works better in SQL

The example app will work fairly well except when it has to deal with large datasets. One should not do browse filtering in ValidateRecord in such a situation. Also counting tags with a loop is not a good idea if there are many tags. I originally developed this tagging method in SQL, and there you could do all the filtering using joins. I also created a view of the tag file to replace the TagCount procedure.

Another feature of SQL is the ability to change some of the Tag procedures to

pass a `PROP:Sql` statement to the back end to accomplish its task – especially deleting all the tags for a specific tag area.

But I wanted to do it differently!

In the app I have a call to `TagIsTagged` for each row in my browse to determine if it is tagged or not (see the icon section in `BrowseMembers`). This is also not efficient. In the original SQL app I used a custom join. See the `BrowseMembersCJ` procedure to see how I had it set up using SQL. This was neat as I could add more custom joins to aliases of the `UserTag` file to retrieve the tagging status of the various attributes of a member. In other words, I could custom join to `AliasUserTags` on `user`, `EQTA:MemberTypes` and the type in the member table, thereby giving me access to the member type tagging status for each row in the view..

Also if I made the `UTG:TagValue` field hot I could make my show tags filter work on the column which would pass the filtering to the view instead of testing each line of the returned result set in the `ValidateRecord` method.

Unfortunately I could not get this technique to work reliably with the `TopSpeed` driver. Notice the strange behaviour of the tagging and icons in `BrowseMembersOld`. It seems that the custom join produces strange results. I am probably missing something basic and would welcome a solution, as I would much prefer to use a custom join. So if any of you can figure out what I am doing wrong, please show me.

Stay tuned for the next exciting episode

During all of this I have realised that there are other ingenious ways to use this approach to tagging. I can store whatever I want in the tagging table. The most obvious is the ability to pass data between procedures or even applications without needing to hardcode the passing structure. I can also use this method to store variables (like filter parameters) per user instead of using INI files and global variables. But that is a subject for another article.

[Download the source](#)

[Chris Laurie](#) hails from Cape Town, South Africa and has been into many things not always IT related, from the stock exchange to performing in garage bands. He has been programming as a hobby since the 1980s, and for a living (with Clarion) for the past 10 years. He loves to cook, drink wine and go for long walks in the forests in Table mountain. Since Dec 2003 he is also a proud and happy farther.

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Reborn Free**CLARION**
online

Handling Nullable Fields in an ABC Application

by **Harley Jones**

Published 2004-09-17

It's no secret that Clarion doesn't handle nullable fields very well. In keeping with the ideal of simplicity, Clarion considers the value of a null string to be blank, and a null number to be zero. And in most cases, this is fine, and can make your code easier to use and maintain. But there are occasions when you have a nullable field that really needs to be null.

Clarion provides some special functions for dealing with null values. For reference, I'll point you to `NULL()`, `SETNULL()`, and `SETNONNULL()`. But if Clarion has the ability to play nice with null values, why doesn't it? Because the dictionary doesn't distinguish a nullable field from a non-nullable field.

Enter the User Options feature. Every field in your dictionary has a potential collection of User Options. The IDE gives you a nice interface for defining User Options and specifying their values. Behind the scenes, they are stored as a collection of concatenated strings. So, what are User Options? Put bluntly, they are a template-only collection of name-value pairs.

For every nullable field in my dictionary, I've added a User Option called "Nullable," and set its value to "True." To do that, select the Options tab on the Column Properties screen. Right-click the listbox and select Insert. Place "Nullable" in the Label textbox, and select Boolean as the datatype. Figure 1 shows how it looks in the IDE.

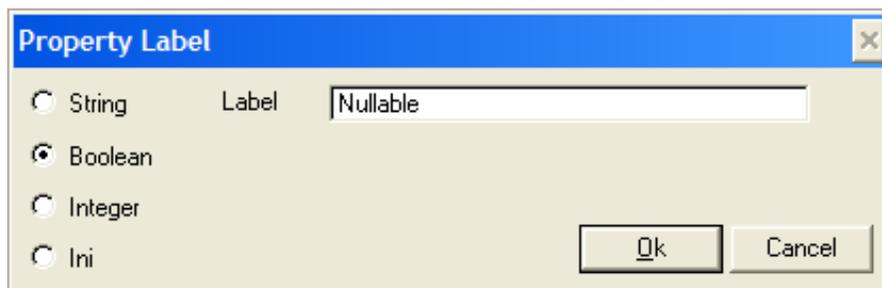
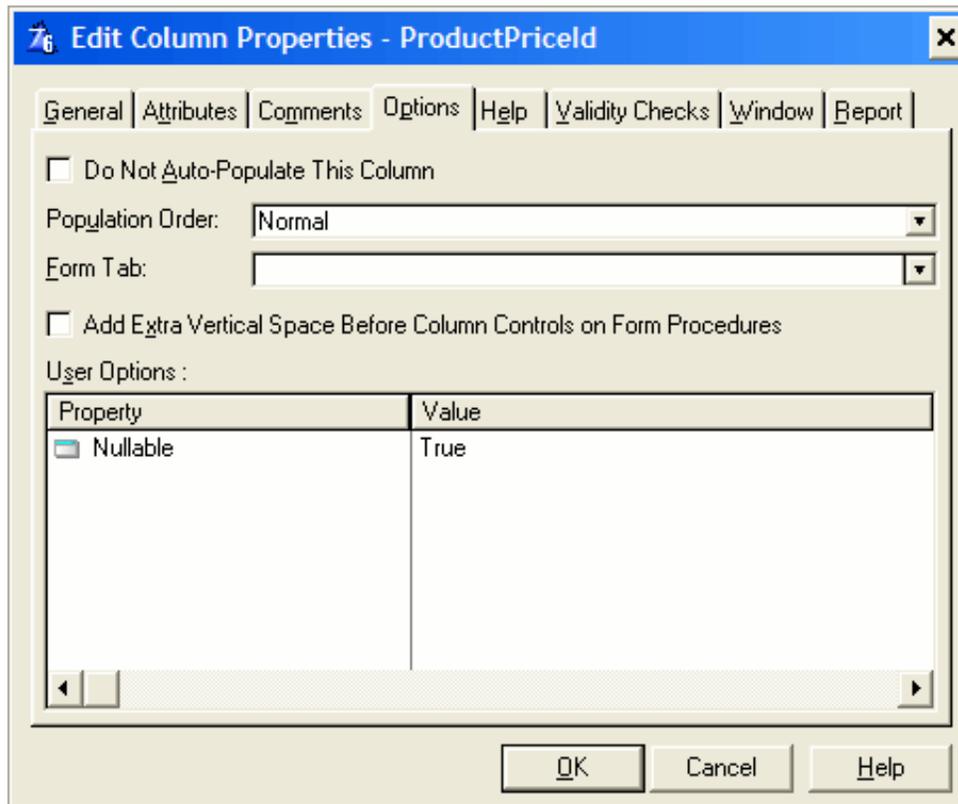


Figure 1. Setting the user option's label and data type

Click OK, and change the value of Nullable to True by selecting the row, and checking the Edit-In-Place checkbox, as in Figure 2.

**Figure 2. Editing Nullable's setting**

If you export the dictionary to text, you'll see this:

```
[USEROPTION]
!Nullable(1):Boolean
```

Remember, this value is template-only, and I just created it. So there is no template using it yet, and no code is being generated to handle the new value. I need a template.

My immediate problem is that I use Guids in my SQL database as the ID for every record. And all Foreign Keys are based on these Guids. But sometimes, a foreign key is optional. Well, a Guid can't be blank, but it can be null. I first noticed this problem when attempting to insert a record with some blank foreign keys. An error message popped up stating that a value could not be found in the foreign table. This was obviously true, because I didn't specify anything.

My application does this lookup because I've defined an "InFile" validation check for this field. In other words, "If there is a value in this field, make sure it's a valid value before writing to the database." And that's exactly what the application will do, using the standard template-generated code. But now with my Nullable User Option, I can tell

Clarion to skip the checking if the value is blank. I do this with the following template code in an application extension (see the downloadable source code for the template and an example of its use):

```
#AT(%FieldLevelValidation),WHERE(%FieldLookup AND (%FieldValidation =
'INFILE')),PRIORITY(5999)
  #IF (INSTRING('Nullable(1):Boolean', %FieldUserOptions, 1, 1))
IF %Field <> %(CHOOSE(INSTRING('STRING', %FieldType, 1, 1) = 0, '0', '')) THEN
  #ENDIF
#ENDAT
#AT(%FieldLevelValidation),WHERE(%FieldLookup AND (%FieldValidation =
'INFILE')),PRIORITY(9001)
  #IF (INSTRING('Nullable(1):Boolean', %FieldUserOptions, 1, 1))
END
  #ENDIF
#ENDAT
```

This template generates two lines of code around the validation code, similar to this:

```
IF Fil:ForeignId <> '' THEN
! standard validation code
END
```

Clarion is already generating the validation code, which falls inside the IF statement. Now that pesky error message no longer appears for blank foreign keys. Of course, Clarion will still try to write a blank to the SQL backend. SQL will then try to convert the blank string to a Guid, and fail. To fix this, I use another short template:

```
#AT(%FileManagerCodeSection, , 'TimeStamp', '()), PRIORITY(7500)
  #FOR(%Field), WHERE(%FieldType <>'BLOB' AND INSTRING('Nullable(1):Boolean',
%FieldUserOptions, 1, 1))
IF NOT NULL(%Field) AND %Field = %(CHOOSE(INSTRING('STRING', %FieldType, 1, 1) = 0, '0',
''))
  SETNULL(%Field)
END
  #ENDFOR
#ENDAT
```

If you can read the template code, and you're familiar with the ABC FileManager, you're probably wondering where the TimeStamp method came from. You'll need the code this template generates whenever Clarion attempts to write to the backend. For plain ABC coding, this means six places: Insert(), PrimeAutoInc(), TryInsert(), TryPrimeAutoInc(), TryUpdate(), and Update(). I've derived the FileManager class and added my own TimeStamp method for time stamping each record before it's written to the database. I then derived the six ABC methods which write to the database so they call TimeStamp first. I suppose client-side triggers could limit this to two locations: PreInsert and PreUpdate. In any case, this code simply sets the field to a null value if it is blank.

And that's it. Hopefully, I've shown a useful solution to a real-world problem, and I hope

I've opened a few people's eyes to the power of User Options and templates. Attached is my FileManager class, and a template file which can be included in your own template set to implement the workarounds detailed in this article.

[Download the source](#)

[Harley Jones](#) graduated from a math and science school, so he could go to a liberal arts college and get an English degree, so he could become a programmer. He listens to loud music while he works, likes to eat gummi savers, and is always looking for ways to program more by programming less. He lives in Mobile, Alabama, where he spends all his money on his wife, daughter, and the occasional comic book.

Reader Comments

[Add a comment](#)

Excellent Harley, I've needed this for quite some time....

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)



DevCon 2004 Starts Off In Style

Published 2004-09-19

DevCons are like weddings – there always has to be something that doesn't go quite according to plan. If everything goes too smoothly, you don't have any good stories to tell.

I figured I was taking a bit of a chance flying in to Orlando late Sunday afternoon. The reception, at the Marina in front of Disney's Contemporary Resort, was scheduled to run from 6:00 p.m. to 9:00 p.m., and my flight was scheduled to arrive right at 6. Assuming my flights were on time, I figured I'd dash for a cab and with luck be schmoozing lakeside by 6:45. And surprisingly, Ted (a division of United – what's next, Air Bob?) did get me into Orlando on time, at least according to the federal definition of "on time", which is to say, within 14 minutes.

The only problem was the huge lineup of disembarking passengers waiting for taxis, evidently due to a highway accident that also kept other, earlier-arriving Clarionites from getting to the hotel in a timely fashion. I was about the thirtieth person in line, and it took a good half hour before I got to the front of the queue. I piled into the waiting conveyance, we shot backwards out of the parking spot, and nearly t-boned another cab coming in for a hot landing. Hard on the brakes, a quick trip, F1 style, over the curb, and we were racing for Disney's Contemporary Resort.



It seemed to be a day for road warriors. Earlier that day SoftVelocity's Doreen Williamson was driving to the conference and got into a serious car accident. From what I've heard, she's okay (and here at the hotel) but it was a big wreck. Of course, Doreen, true to form, wasn't thinking about whether or not she needed to go to the hospital to get checked out. She was too concerned about the conference materials (including all the name tags) that had been thrown out of the car (luckily the tags were still in the box). Her first call wasn't to 911, it was to Bob Z, and even while being loaded onto the stretcher she was giving instructions to helpful bystanders about what needed to be hauled out of the middle of the road.



Thankfully my trip was a lot less eventful. By the time I arrived at the hotel, got checked in, and made my way out to the marina (around 7:30), the efficient staff were already giving notice that the food would be cleaned up shortly.

Having foregone the airline, ah, cuisine, I was ready for some real vittles, and after a few Hi how are ya's I tucked into the mushroom-stuffed ravioli and a helping of fajita fixins. I never got a chance to try the shrimp, and that still bugs me.



It was a fine evening, with the weather a bit cooler than usual (meaning it was very pleasant and not particularly hot or humid), good food and drink, a Key West style guitarist (think Jimmy Buffet), and a chance to do the meet and greet with old friends and new.

And this being DisneyWorld, we were treated to the nightly fireworks display.



Reader Comments

[Add a comment](#)

Hi Everyone Having a great time, beer is great and...

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)



DevCon 2004 Monday Keynote: Bob Zaunere (Part 1)

Published 2004-09-20

Bob Zaunere, SoftVelocity's president, gave the keynote address on Monday morning. After welcoming the attendees to the hurricane zone, and pointing out the exhibit hall and the bathrooms, he set the promised conference tone with a complete focus on Clarion technology, and a total absence of "marketing hype." DevCon 2004, said Bob Z, has two purposes: to show everyone where SV is headed, and to get feedback from developers about where SV needs to do better.

Bob then launched into a summary of the new technologies on the way, some of which will be for C6, and some of which are part of C7. In the former category, the IP driver is about two to four weeks away from release. It is code complete, but is waiting on documentation (not the fault of the documentation writers – they just haven't yet been given what they need). It should be a ten minute conversion to make any app work over an IP connection. What this means is that you can now separate the Clarion application from its database via an IP protocol connection (without, for instance, having to resort to an SQL database). The IP driver is a socket-based way to connect to your back end data. It is not, however, made to replace a back end – it has other abilities. It is well suited for information exchange between clients, or intermittently-connected workers.

The Dynamic File Driver can be used with ISAM and SQL drivers to

create tables on the fly. You're not bound to the dictionary. This driver is particularly well suited to SQL back ends, since you can create records that exactly match data returned from a stored procedure or SQL statement.

The new PHP templates let you generate complete PHP web apps, using a similar approach to the ASP templates. Pierre gave a presentation later in the morning – look for coverage in ClarionMag in the coming days.

Most of the rest of Bob's presentation dealt with Clarion 7. First off were a number of editor improvements. I should note that didn't actually see a live 32 bit IDE running – instead, Bob showed a number of screen shots, which were also reproduced in the conference documentation.

Clarion 7 design goal wasn't simply to move the IDE to 32 bit, but to improve upon it. I'll report on this in more detail in Part 2, but in essence the work on the IDE focuses on making it more customizable, and in particular “flattening” the interface. That means that instead of having to drill down from one dialog to another, as is now the case, you'll be able to get at a lot more information with just a click or two. Tabs are a much more important part of this IDE.

The new editor is also showing signs of a major rewrite, with features including an Intellisense-like feature, full macros, code outlining/collapsing, and (at last!) multi-level undo capability. As well, the editor will be configured for C++ as well as Clarion, and apparently Bob will talk more on Tuesday about other supported languages, including C#. Something to think about...

That's the teaser. I need to get back to the sessions, so stay tuned for [more](#) on the Monday keynote in the next installment. And Tuesday promises to be a very interesting day, as Bob will be discussing Clarion.NET.

[Read Part 2](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)

Reborn Free

CLARION
online

DevCon 2004 Monday Keynote: Bob Zaunere (Part 2)

Published 2004-09-21

As I indicated in [Part 1](#) of this report, Bob Zaunere's Monday keynote covered upcoming Clarion 6.1 compatible products, such as the new In Memory, IP, and Dynamic file drivers. Most of Bob Z's time, however, was spent on the feature set for Clarion 7.

First of all, yes, the C7 IDE will be 32 bit, with much of the code being rewritten from the ground up. Based on what Bob showed, I expect C7 will still look quite familiar – many of the changes are under the hood. But there are quite a number of significant IDE improvements as well.

The source editor gets a major makeover in C7. It is a multi-language editor, able to handle C++, C# as well as Clarion. New features include call tips, which are tooltip-like windows that show function call information (in the same format as the help file) after you type the name of the function. Along with call tips the editor has code completion – for instance, if you type the name of a known object and then the period, the editor will let you pick from a list of available methods. You can also “fold” or outline code which is in a structure. In C++ code, the editor can collapse code inside the { } braces. In Clarion, structures completed by END statements can be collapsed or expanded.

And at long last, there will be a multiple level undo feature. That alone would probably be enough to make me upgrade. There are

various other improvements like autosave capability and language-specific color schemes, and you will be able to open one file multiple times, and have the changes in each editor window merged back into the file.

Probably the most visible change in the IDE is the flattening of the user interface, both in the AppGen and the Dictionary Editor. You will see more tabs, meaning you won't have to drill down from one window to another to set individual properties. And in the window and report formatters you will be able to create a custom view of commonly used properties to help speed up work

The dictionary editor will have global search/replace features, so for instance you can make changes to multiple fields/tables without having to export to a TXA. Other dictionary features include table groups, for better organization of large dictionaries, system tables, which are aggregates of everything in the dictionary, sorted any way you like, better dictionary reporting, and a synchronizer that doesn't lose all your work when it encounters an error. Under consideration is the possibility of offering remote access to the dictionary, using the IP driver.

A new SQL query tool will be available from within the AppGen, the dictionary, and the source editor. This tool was originally developed for the J2EE templates (which never shipped).

A new project feature, called the Clarion Solution file, lets you build multiple projects as one task. You can combine .PR, .PRJ, and .APP files, and (unlike VS.NET) you can mix and match debug settings for the different projects.

Bob Z mentioned three potential template system improvements: a template debugger (there used to be an internal template debugger in the CFD days); a template dialog editor; and compiled templates (seen as an advantage for third party vendors). The first two at least received a warm reception. ([play the video](#) - if you have problems, try doing a right-click and choose Save As to download first before

playing)

The ABC and Clarion template chains will both get some work, but in the form of enhancements, not radical structural changes. In other words, the focus is on making the upgrade from C6 as painless as possible.

The RTL also gets a tuneup in C7. Improvements include XP theme support for remaining controls, although the list/browse box will probably never get complete theme support because Clarion's list control is so different from Microsoft's list control. PNGs and TIFFs will be supported.

There are only small file driver changes – at this point, there isn't much improvement to be had. A class API to the SQL drivers is about the only significant possibility, and will probably happen, but it all depends on developer feedback.

A number of other issues were raised in the Q&A session. In C7 you will be able to have any number of apps open at once. The IP driver was discussed, and Bob again pointed out that this is not intended as an SQL replacement. It is useful for TPS access on a LAN; since actual table access is being done by just the server, there are no oplocks problems. Asked about stored procedures, Bob pointed out that eventually you will be able to write these in Clarion.NET as well as import them from the SQL server. On Linux, Bob said the way to approach this is via Clarion.NET, since there is a lot of work being done to port .NET to other platforms.

Reader Comments

[Add a comment](#)

One of the planned improvements for the file drivers is...

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

online sales and delivery
for your applications & tools

Developer **PLUS**

DevCon 2004 Keynote: Clarion.NET

Published 2004-09-21

Tuesday's keynote address by Bob Zauere was titled "From Win32 to .NET", and gave the attendees a look into the not-very-distant future. And the future is Clarion.NET. To make the point, all SV staff showed up at the session wearing their snazzy new Clarion.NET shirts!

Bob began with an unapologetic lack of graphics, and a list ten top Why .NET? reasons:

10. .NET is cool. It's really cool. Ask Microsoft, or get another opinion from Sun.
9. You build better apps. Make a .NET app, and a Win32 app, and you know the .NET app is better. (And after this, Bob actually got serious about his Top 8 list.)
8. Consistency: With .NET, Clarion developers finally have a single library for Windows, and .NET can run on Windows versions all the way back to Windows 98.
7. Programmer adoption: There have reportedly been 90 million downloads of the .NET SDK, and there are predictions that 80% of all developers will be using a .NET language in two years.
6. Rapid development: .NET promises faster development; Clarion



will add at least 25-50% more productivity on top of .NET's gains.

5. **Platforms and devices:** Microsoft is expected to release a 64 bit .NET framework this year, which means you can write one application that runs on 32 bit and 64 bit machines. .NET also opens the door to Linux, Mac OS/X Solaris, and other platforms (via the Mono project and the DOTGNU Portable .NET project), as well as mobile systems.
4. **The Borg (a.k.a. Microsoft):** Planning for .NET is prudent, and the logical choice. Microsoft and .NET will be around for a long time.
3. **Web apps:** ASP.NET is a powerful tool for building web applications, and with Clarion.NET you will be able to build those apps using the familiar Clarion language.
2. **The languages:** All .NET languages are interoperable, but each language has its own strength. Clarion's assets in a .NET environment are its data access grammar and its language syntax, not to mention the template language.
1. **Preserve existing code investments:** There are millions of lines of Clarion code out there. Clarion.NET will let you keep that code. Bob also pointed out that each month SV upgrades about a dozen DOS developers to Windows.

There are lots of terms you will need to know:

Longhorn is coming, even though it will have backward compatibility. This is the next version of Windows after XP. It is not a complete rewrite of the OS, and it will have Win32 APIs. But at least part of the Win32 API will be labeled as "legacy." Of course sometimes new versions of Windows break old APIs.

WinFX (FX as in special effects) is a superset of the .NET framework, and which exposes new Longhorn functionality. WinFX works with managed code, and coexists with Win32 but Win32 will be a legacy interface.

Avalon is MS's XML presentation platform, and SoftVelocity may adopt the Avalon XML specification in defining Clarion window structures. MS recently announced plans to backport Avalon to Windows Server 2003.

Yukon is the next generation MS SQL Server, and among other things it integrates the .NET CLR with the database. Among other things this lets you write stored procedures with .NET languages (such as Clarion.NET!).

Clarion.NET design goals

Bob stated SoftVelocity's design goals for Clarion.NET are to create a language that has the ability to interact fully with the .NET runtime and other .NET languages (audio, 57 seconds: MP3 or WMA) . He then gave some definitions related to the Common Language Specification (CLS):

- A CLS Consumer is a compiler that can emit code that can consume all of the features of the CLS.
- A CLS Producer is a compiler that guarantees all of its output is CLS compliant
- A Full Producer – like Clarion.NET – is a compiler that guarantees all of its output is CLS compliant, and defines mechanisms for generating every feature of the CLS.

Audio: Listen to Bob Z's comments on interoperability - [MP3](#) or [WMA](#)

Clarion.NET will add some language extensions, and some constructs may be deprecated.

Bob pointed out that Clarion is already a good match for .NET. That's not the case for all .NET languages. For instance, the migration from VB to VB.NET has been a huge headache, and you will read recommendations that applications simply not be ported – it's better to do new development.

On the other hand, you could produce managed Clarion.NET code right now if you had the compiler (you don't, Bob does). That term

managed is important.

Managed code is code that is supervised by the runtime environment, to make sure it behaves as expected. Managed code benefits include:

- Memory management and garbage collection (which is becoming more of an issue for Clarion developers these days)
- Runtime checking – exceptions are thrown if you write past the end of a buffer or use a null pointer
- Security – the Framework Class Library (FCL) and the CLR have a strong security model
- The end of the GPF? Not quite, but exception handling can avoid crashes

Also important to note is that procedural code will still execute, even though under the hood everything is class based. You can create Clarion.NET apps without using OOP at all, if you wish.

Bob gave some examples of new language features in Clarion.NET. Exception handling uses the TRY, CATCH, and FINALLY, familiar to anyone who has done Java or C# development. New data types include bool, char, and datetime. All existing Clarion data types are still supported. Other additions include properties, events, delegates, and indexers. New statements include: PARAMS; CHECKED/UNCHECKED; FOREACH; NAMESPACE; USING. Clarion.NET adapted syntax used in C#, since that language is more like Clarion than VB, and SV likes C# better than VB.

Clarion.NET code can look exactly like the Clarion code you write now, as one of Bob's examples demonstrated. That program didn't use any .NET-specific features, but it did show how existing code could easily run as a .NET application. You may need to make some changes to your application to have it run as managed code, however.

Notes on the .NET architecture

Compiling a .NET application is a two step process. First, the high

level (i.e. Clarion, C#) code is compiled to an intermediate language (IL). This code looks more like machine code than high level code, but some people do like to write it. The IL code is packaged into either an EXE or a DLL, and the second compile takes place at runtime by the JIT compiler. Only used code is JIT compiled - if a method isn't called, it isn't compiled.

The IL code is what makes a .NET application portable, and this can have some positive performance implications, because JIT compilers can optimize code for the particular CPU in use. Traditional compilers have to target the lowest common denominator.

There is also a utility called Ngen (Native Image Generator) which will let you precompile your application. WinForms is an example of precompiled .NET code. You can also run Ngen as part of your application's install process, so that your code is pre-compiled and optimized for that machine.

Although managed code is an important feature, .NET also provides a way to create unmanaged code. For instance, Win32 code is unmanaged and "unsafe" in .NET terms.

You might also want to use unmanaged code if you want to use the .NET class methods directly, i.e. ADO.NET, or if you want to call an unmanaged driver (say Btrieve API) directly.

In the Q&A session, Bob Z pointed out that there will be a private group for developers to give their feedback on Clarion.NET. SV also wants to include the third party vendors, and they will have a private newsgroup as well (immediately, in Bob's words). While these groups will be private, they will probably not be restricted to DevCon attendees. There is also a good possibility of an EA program for Clarion.NET.

It's expected that the compiler will be finished before Christmas, and the RTL work can't be completed until the compiler is finished - work on both is under way. Bob wouldn't commit to a timeframe for deploying Clarion.NET applications. He did say a working environment could be available fairly quickly (presumably that means after the

compiler and RTL are done).

Again, this is a separate product from C7. That raised the question of whether Clarion developers would be faced with two code bases to maintain. Bob allowed some developers might feel they needed to keep Win32 and .NET versions of their applications, but he reminded us that .NET is available for Windows platforms all the way back to Windows 98, so you might find that you simply want to upgrade to .NET. And that brought up the question of whether there would be a Clarion 8. Most likely yes, because Win32 isn't going to go away overnight, and there will be developers who choose to stay with Win32 rather than moving everything over to .NET.

It's undetermined if the list box will become the WinForms data grid but that's likely.

SV intends to create a Clarion.NET plugin for Visual Studio. This means the AppGen, dictionary editor, source editor (most likely) and compiler would be available in the context of Visual Studio. After some initial confusion, there was actually quite a lot of interest in this, but Bob contended that it wasn't that important an issue, since the only people who would really benefit from this arrangement would be those who were already using Visual Studio for other .NET development.

Bob asked for a show of hands of those who would buy Clarion.NET as soon as it's available. Most raised their hand, and most also responded to the question of whether they would want to use Clarion.NET with an ISAM file system (such as TPS files). Many were also interested in some early training in .NET.

SV's new ASP.NET product, which uses Clarion.NET, could be released fairly quickly, since that doesn't need all of the window handling code (i.e. Avalon, WinForms).

Reader Comments

[Add a comment](#)

Great article on some of the most exciting news we have...
Yes, I agree with Kevin. This is very exciting news and,...
Sounds like Clarion's about to hit the mainstream! As...
Awsome! Very simply awesome!
Rhys, Porting a Clarion app to .NET will not...
Hurray! Dave Eggen owes me a beer! Remember that...

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)



DevCon Pre-Conference Training: Day 1 & 2

by Gregory Bailey

Published 2004-09-23

With more than 140 developers in attendance this wasn't your usual training class. In fact it was more of a guided tour of the new features of Clarion 6 than training.

Materials handed out to all in attendance included a CD with all of the example applications and projects that were used to demonstrate the new features, and a hand book with 280+ pages of information about the topics to be presented. Additional materials that hadn't made it onto the CD were promised to all.

Bob Foreman and Robert Rickets (funny how many Bobs there are at SoftVelocity) were the trainers and did a remarkable job demonstrating the materials on the two very large screens. The first day they covered new features in the Dictionary and Application Generator, along with Wizard Themes, Version Resources, XP Manifest support, and rebasing multi-DLL applications.

A quick poll of the attendees showed that most of the 140 had been using Clarion since before 1995 and 25-35 had been using Clarion since before 1990 (including me).

Features in the Dictionaries that were covered included the Synchronizer, new file conversion techniques and client side triggers.

New methods of working with browses were covered next, including end user configurable browse format strings, browse styles, edit in place and the query center. Forms improvements included using enter as tab to advance from control to control and visual indicators to show which control currently has focus. VCR forms were also covered.

Report improvements include the ability to print from queues, dynamic sorting of reports and a new break manager that allows you to print headers and footer yourself, and get in before printing footers to allow calculated fields.

Day 2 moved on to more complex subjects. The project system was covered along with an explanation of what the differences are between the threading models used in C55 and C60. All of the methods and templates available to change the threading model were discussed (as a preview to the breakout session to be given by Steven Parker later during the conference).

Business graphing was covered, showing the basics of setting up the graphics template and the myriad of configuration options available to choose different types of graphs and methods of grouping information points in the graphs that are now available. One very interesting aspect of graphing available is the ability to drill down to the data that supports the graph, with a click of a mouse.

HTML help setup and use were discussed along with the messaging templates. Also included were techniques that need to be used to set up the messaging templates more easily for the end user.

The new In Memory Database Driver (IMDD) was demonstrated, and appears to be a very useful addition to any Clarion programmers' toolbox. The IMDD can be used any place that a normal file can be used, and relationships can be created with the IMDD and any other file type to make views that are available in a browse or a report.

The Dynamic File Driver (DFD) was used to show how a file can be

easily created on an ad hoc basis. A lively discussion ensued in attempting to figure out how to use this driver to import a file with an unknown format. This was left as something to try at home. The DFD is scheduled to be offered for sale this month (September).

The Internet Protocol (IPD) driver was used to show how you can access any file type at a location identified by an IP address. A JAVA applet (included with the driver) can be used to configure which files are available and who can access them. It is unclear how much information will be sent between the client and the server, and also more importantly how much information will be sent from the server and the client. This does look like a very promising technology. It can be used across the internet or across the LAN or even locally on your LAN. The IPD should be on sale sometime in October.

Reader Comments

[Add a comment](#)

**The IP Driver is on sale now. The price is 300.00. I got...
I got mine as well. SoftVelocity was evidently...**

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine



DevCon 2004: Developing Applications with the new IP, Memory and Dynamic File Drivers

by Tom Hebenstreit

Published 2004-09-23

This session was delivered by the inimitable Bob Foreman of [SoftVelocity](#), and was intended as an introduction to three new and/or recently announced add-ons for Clarion.

In-Memory Database Driver (IMDD)

The In-Memory Database Driver is a new driver that is used the same way as, and behaves exactly like, any other file driver. The trick with this driver is that there is never an actual physical file – all inserts, changes and deletes take place in memory. Think of it as a queue that uses standard file syntax, has a FileManager, and that can be used like any other file in your dictionary. This means that, unlike a queue, it can easily be used with any template that requires a file. Examples of use include things such as browses, processes, reports and many others.

Things to note:

- IMDD tables are transient, and only exist while your program is running. If you need to save IMDD data between sessions, you will need to write that code yourself.
- Performance is excellent since there is no disk I/O taking place. The exception would be if you tried to load so many records into

an IMDD file that you used up all physical memory and overflowed into virtual (i.e., disk) memory. Like queues, IMDD records are compressed before being written to memory, so memory usage should be reasonably efficient.

- IMDD files have built-in thread synchronization, so they are inherently thread safe. If you use global non-threaded queues, using an IMDD file instead will ensure that you will not run into any thread related corruption issues.
- There is no built-in mechanism for keeping an IMDD table and a physical table in synch. If you need to do that, you will have to write the code yourself.
- One attendee asked if there was a way for multiple programs running on the same machine to share an IMDD table. The answer was no, an IMDD table is local to the application that created it.

The In-Memory driver is available now and lists at \$225 US.

Dynamic File Driver (DFD)

This one is actually a bit misnamed, in that it is not really a file driver. It does not appear in your list of registered drivers, and you cannot assign it as the driver in the data dictionary. What it does is let you dynamically define all of the properties of a file at runtime using PROP syntax, such as the driver, name, keys, fields and all the rest of the details that are normally generated from your data dictionary. Once you have specified the properties of the file, you use a new language statement ('FixFormat') to "lock" in the definition, create the buffer, etc., and attach your new file to a real driver such as the TopSpeed, ASCII or BASIC drivers. From that point, you can use all Clarion language statements and/or driver SEND strings that are valid for the driver you specified.

A simple dynamic file declaration for an ASCII file with one field might look like this:

```
MyFile      &File      ! A reference to a file
```

And then in code:

```

MyFile &= NEW(FILE)           ! Create a new instance
MyFile{PROP:Driver} = 'ASCII' ! The real file driver
MyFile{PROP:Create} = True    ! Can create physical file
MyFile{PROP:Type,1} = 'STRING' ! Declare a string field
MyFile{PROP:Size} = 255      ! ...of this length
FixFormat(MyFile)           ! Lock in the definition
! ...do stuff with the file...
UnFixFormat(MyFile)         ! Free up/discard the dynamic file

```

Things to note:

- One use mentioned was to use a dynamic file to load the results from an SQL query into a matching file.
- You can use the In-Memory file driver with dynamic files, meaning you can basically create queues on the fly. It also gives you the capability to create the equivalent of ADO disconnected recordsets, where you load the results of a query into memory and work with the records without needing a constant connection to a database.
- Questions were asked about having the capability to read header information from a file and use it to build a matching DFD definition (similar to TopScan.) The short response was no, it won't do that. It is possible to do it yourself if you write code to read and parse a data file header, but you would also need to know the exact header format of the type of file you want to read. For TPS files, that information is proprietary to SoftVelocity.

The Dynamic File Driver will be available by the end of September, and the price is slated to be around that of the In-Memory driver.

Internet Protocol Driver (IPD)

The Internet Protocol Driver basically allows your programs to connect to another server component for their file access. Think of it as another layer of code inserted between your program and a standard file driver. This layer takes the file access requests such as GET, NEXT or PUT from your client program, transfers them to the server using IP, and then that component performs the actual physical file access and returns the result.

How does it work? In your dictionary you would define your file as usual, choosing whatever file driver the physical file will use, e.g., TopSpeed. A special new flag must then be set on the File Properties User Options tab to flag the file as one that will be using the IPD.

New templates are provided help you to create the special server side components that will handle IPD requests. These components are simple Clarion applications that pretty much just need to be created and compiled. Global extension templates are also provided for the client side that let you enter the connection information so the client can locate and connect to the server components. Additionally, a Java app is included that let's you remotely manage the server components of the IP driver from any location.

Things to note:

- IP and local files can be mixed in the same dictionary and programs. For those files that have the special flag, extra code will be generated and/or functions linked in to implement the IPD functionality.
- The client and server components of the IP driver can both run on the same machine.
- If your program cannot connect to the IP data source (say, your laptop doesn't have an active connection to the Internet), it is possible to detect that in code and switch to using a local version of the file. This is not automatic, but can be done with embed code.
- When asked about the performance impact of using the IP driver on a LAN, Bob Z. said that some things can be faster, some slower and that in most cases it was pretty much equivalent. If you venture beyond a LAN to the Internet, then performance will obviously depend on connection speeds, traffic congestion, etc.
- The server component requires NT/2000/XP in order to run as a service, but it can also run as an executable under Windows 9x/ME.
- Questions were asked about data and connection security. The answers were not really clear, but it seems that the driver will support the use of SSL (Secure Sockets Layer) for the connection, but data is not encrypted beyond that. User ID's and passwords are also used to help manage attempts to connect to

an IPD server.

The IP Driver is scheduled for release in October 2004, but the pricing has not been determined yet.

A longtime Clarion user, [Tom Hebenstreit](#) is an admitted tool junkie who refuses to go straight and code without his arsenal of third party products. During those rare moments when he isn't either using or writing about Clarion, he indulges his twin passions for blues and beer by performing around Southern California in a variety of totally-obscure-but-famous-any-day-now rock and blues bands.

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)

Reborn Free

CLARION
online

DevCon 2004: Clarion In The Public Sector

by Stephen D. Gradolph

Published 2004-09-29

This presentation by Douglass Smith may have been the sleeper hit of DevCon 2004. Attendees were so disappointed at missing the beginning of his presentation that Bob Campbell (who had the following session) graciously yielded part of his time for an encore performance. Attendees were greatly interested in Doug's groundbreaking work in platform independent remote computing, for which Doug modestly gave much of the credit to Clarion. Doug also shared some of his perspectives on working with governmental agencies.

Doug started with some tentative answers to the question: Why is Clarion a good fit for the public sector?

These included a list of Clarion's strengths familiar to DevCon attendees, but Doug also listed the equally well known hurdles to wider adoption. Some background followed before presenting two specific case histories of Clarion success stories. To wrap up, Doug presented some of the lessons learned in using Clarion for a governmental agency, including positive suggestions for making a successful case for Clarion. Most questions were about using Clarion to solve web and connectivity problems, so the reprise focused on the technology.

To put the case histories into context, Doug continued with some background about the mission and culture of his employer, the

Kansas Department of Agriculture or KDA. With about 400 employees in the department, nearly half the users he supports never come in from the field. The crazy quilt information technology patchwork includes Oracle, MS SQL Server, MS Access, and several others database back ends. The desktop platforms include Windows XP, but attendees really perked up when he mentioned TabletPC, PocketPC, and MS Smartphones. Attendees were surprised that his department only has five programmers, but were able to guess that only one used Clarion.

In the first case history, Doug had only 90 days to meet a completely new State-mandated reporting requirement. Now his department would have to track vehicle costs, mileage, and maintenance for their entire automotive fleet, and generate the same blizzard of paperwork the State had produced previously. Nor could Doug start from scratch because his new application would have to communicate with cantankerous computer systems the State or his department continued to use. Besides the incredibly tight deadline, Doug also had to address newly important security concerns and keep an eye to future flexibility and expandability. Attendees were so impressed with sample screen shots that Doug could probably have spent a lot more time on this one application, but the real story was no news: Clarion, with help from key third party tools, made success possible.

Attendees really got excited about the second case history. Other new requirements created an enormous problem that few organizations could solve with any tool or the most generous resources. To further complicate reporting challenges, which Doug compared to filing IRS forms, remote employees needed the ability to file inspection reports remotely, and summary results were to be reported via the Internet as well. Doug is keeping his fingers crossed because he still has two weeks to deadline, but he was able to demonstrate some amazing technology. Once again, Doug showed how this difficult challenge was met by Clarion and third-party tools.

Of all the third-party tools Doug cited, two loomed largest. First Doug described how the Fenix development platform provided by

RADventure helped him create an entirely new HTML-based solution for remote connection via PDAs. Doug explained how Clarion's window formatter painted a fairly accurate preview of how the Fenix produced HTML forms would be rendered; samples of simply elegant windows proved his point. For further proof, Doug took attendees to the KDA website, www.ksda.gov, to show some rich QBE forms and handsome SQL query results hosted within a DotNetNuke frame. Behind the scenes, Doug explained how he applied the database replication templates provided by CapeSoft to solve the problem caused by remote users with disconnected access to local copies of the database. Doug also gave credit to the IceTips SQL templates and three other CapeSoft tools: FM3, NetTalk, and SecWin.

Finally, Doug shared some lessons he had learned with respect to the challenges facing software developers in large organizations and the familiar problem of gaining acceptance for Clarion in the larger information technology world. As a true Clarion evangelist, Doug believes early adoption best plays to Clarion's RAD strength by very quickly giving users something concrete and positive to discuss. Attendees seconded the thought and many asked questions about remote computing and related challenges. The subject of discussion might have been any other cutting edge technology development environment, a description many attendees might previously have been surprised to see applied to a government agency in a small mid-western state!

Reader Comments

[Add a comment](#)

It was to bad he was pulled off the stage and probally...

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Reborn Free**CLARION**
online

DevCon 2004: Using Objects in Clarion

by **Hyrum Tatton**

Published 2004-09-29

For those of you who have never heard Bruce Johnson explain OOP and virtual methods, you have missed one of the truly enlightening experiences for a Clarion programmer. I first heard Bruce at ETC III several years ago. His explanation of the ABC classes was the first time I really understood how classes were used by the IDE and the template system. Bruce has the unique ability to get to the heart of the subject in a way that is clear and enjoyable.

After a brief introduction to the language of OOP, Bruce again explained the magic of virtual methods. Virtual methods allow the ABC Classes to call *my* code. In other words if the method is virtual, I can use an embed point to enter code that is then called by the ABC class. I can write an embed that prepares data for the ABC Class Method (Before the Parent Call), or that does some cleanup after the ABC Method (After the Parent Call) or that runs instead of the ABC Method (Before the Parent Call followed by a line that says RETURN).

CapeSoft, where Bruce works (and is a part owner), has provided a free template called [Object Writer](#) to assist you. Bruce used Object Writer to demonstrate writing several classes.

As an example Bruce created a derived popup class. This class was then used to demonstrate the techniques to embed code in both the class definition (properties) and the class procedures (methods).

The second example was a derived IniManager class. Object Writer has the ability to add new methods, or to write a class from scratch. The editor section in Object Writer has one big advantage over hand coding; Object Writer generates the correct structure in the correct place.

Bruce ended with a series of tips. Some are so good that they need to be shared with readers who weren't at DevCon.

1. Use the following syntax to instantiate an object

```
MyClass Class(whatever)
    End
```

Rather than

```
MyClass whatever
```

(This can avoid some funny GPF's)

2. Routines are in scope when you derive a local class in a procedure.
3. Using classes requires good documentation and (hopefully) examples.
4. You can suppress the call to the Parent by adding a return before the call to Parent in a virtual method.

As usual, Bruce did a great job.

[Hyrum Tatton](#) began programming in 1969, while working on masters degree in Chemistry. His computer language experience includes Assembly Language, Basic, Fortran, COBOL, RPG, Neat-3, and Clarion since the DOS days. He now works exclusively in Clarion and COBOL. His professional experience includes time as a pilot, a high school chemistry teacher, university instructor, high school principal, and software developer as CEO of [TES-Software](#). Hyrum really doesn't know what he'll do when he grows up, but he's sure it will be exciting. He is passionate about his family, his church, classical music and golf.

Reader Comments

[Add a comment](#)

Thanks for the review Hyrum. One point of clarification...

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)



DevCon 2004: COM Objects In 15 Minutes Or Less

by Mike Hanson

Published 2004-09-30

For a long time now, Clarion users have wanted to step into the world of COM. Clarion's OLE control enabled you to become a COM client, but it always involved compromise, and often didn't work at all. If you wanted to go so far as to be a COM object (i.e. server), then you were just out of luck.

Clarion 6 greatly improved our ability to work as a COM client, but what we didn't realize is that we could create COM objects too! This was the main focus of Jim Kane's presentation. Although the title implied it was easy, what we found out is that there were many elements required. Fortunately, Jim has continued to be a wonderful benefactor to the Clarion community, and he developed a tool that does most of the dirty work for us. His topic focused on the purpose and usage of this tool. Even though he said "COM Objects in 15 Minutes or Less", the session took 4 hours, and it was jam packed with information.

For those who might have felt that COM is yesterday's news, Jim urged that it has a great deal of life left in it. Most of the world still uses it as a standard tool for language interoperability. It also cures many of the problems associated with regular DLLs in disparate platforms.

Jim clarified the differences between Interfaces (a collection of

methods) and Classes (both methods and properties). In the case of a COM object, it is accessible only via an Interface, even though it usually will have properties within. You can call a COM object with only an Interface definition and an address. INCs and CLWs are not necessarily required.

Here are the main requirements involved in generating a COM object:

- A DLL with four special exported procedures
- Registry code to self-register the DLL
- A class factory that can NEW() a CoClass
- A type library resource
- A class that implements a COM interface with some standard COM methods, plus your custom methods
- All your code in the custom methods (while Jim's free tool does the rest)

COM objects and their elements generally have IDs associated with them. These could be UUIDs, GUIDs, IIDs, and CLSIDs. Jim explained which was which, and when you would need them. The structure to define these in your Clarion code is rather obscure, but Jim's utility handles those details. There are also ProgIDs used to identify the main object itself.

All of these various IDs and associated attributes can be found in the Registry. Jim took us for a tour, pointing out important aspects and quirks. He dealt with the issues of COM registration, "DLL Hell" (i.e. two programs on a machine need different versions of the same COM object), and using OleView to peruse the attributes of the COM object.

Your COM DLL *must* include four standard procedures:

DllCanUnloadNow, DllGetClassObject, DllRegisterServer and DllUnregisterserver. The OS calls some of these, your installation program calls others, and some will be called at runtime.

DllRegisterServer is usually called by the installer, but could also be called by your program to enforce the proper COM object registration when your program starts up. Your program will also call

`DllGetClassObject`, to instantiate the COM Object using its class factory.

Your COM object will have at least one `CoClass`, which must at the very least implement the `IUnknown` interface. Of course, you will need more than that to do anything truly useful. Jim mentioned that `CoClasses` are reference counted when they are used (just like the `FileManager` count calls to `Open` and `Close`, so it knows when everyone is finished with a file). In the same manner, when the `CoClass` reference count gets to zero, it self-destructs.

The more common interface is `IDispatch`, which does everything that `IUnknown` can do, and more. You will likely use this rather than `IUnknown`.

Next Jim went on to explain Dual Interfaces, also known as Early and Late binding. With early binding, your application's references to the COM object are compiled into the program, whereas with late binding, all interactions are handled at run-time. As you would expect, early binding is more efficient at runtime, but sometimes late binding can be easier to implement and is fast enough for the purpose. Microsoft strongly recommends that all COM objects support dual interfaces.

Jim pointed out that the Clarion OLE control could be used with both visual and non-visual COM objects. If your object is non-Visual, you cannot use Events or Licensing. The OLE control uses late binding to access the COM object.

COM objects require a Type Library. This can be linked into your program as a resource (just like a BMP or ICO). The `OleView` utility uses the Type Library to display the information about your object. Jim then went on to show us examples of how various elements like `CoClasses` and `Methods` appear in `OleView`.

To finish the first half of his presentation, Jim showed us how to create a "Hello World" COM object using his free utility. Then he

called it from VB to prove that it worked. He showed us various files that are generated by the utility, as well as pointing out which ones you will change, and which you can safely ignore.

The second half of Jim's presentation expanded upon many of the same aspects as the first, such as Reference Counting, except in more detail. He also described things like QueryInterface, Error Handling & Rich Errors, and COM Events. He ended by create a Clarion-based COM object that could read a TPS file, and return field values for each record to the COM client.

Although Jim's chosen topic was very complex, he did a good job of explaining the key issues, and his COM generator utility should enable many of us to finally start creating COM objects using Clarion 6. Kudos to Jim Kane!

By the way, all of the source code for the utility is provided, so if it is missing something that you need, you can add it yourself. If you do, Jim requested that you send the changes to him, because he likes free stuff too!

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)

online sales and delivery
for your applications & tools

Developer **PLUS**

DevCon 2004: Debugging Clarion Apps

by Geoff Bomford

Published 2004-09-30

Using the Clarion debugger has not always been profitable in my experience, and I was pleased to see, by a show of hands at the beginning of Russ Eggen's presentation, that I was not alone!

A large crowd was in attendance at this breakout session to gain some insight into better ways to debug Clarion apps. Russ began by outlining the most common sources of bugs.

- Errors caused by our code.
- Errors caused by Soft Velocity code in templates and or classes.
- Errors caused by code in third party templates.
- Problems caused by missing, misaligned and corrupt data.
- Problems caused where the code is correct, but the logic and/or design is wrong.

Some programmers claim they don't produce bugs in their code, Russ's response is "They are not coding hard enough!" In other words, bugs are inevitable and tracking down bugs is a process of investigation that everyone will need to face.

Russ reviewed and demonstrated the following tools available to Clarion programmers to rapidly resolve bugs.

- The Clarion Debugger
- A 3rd party API tool – DebugView

- The Clarion driver trace tool

Tools *not* to be used for debugging are `MESSAGE` and `STOP` statements, which are designed for other purposes, can actually impact on program execution, and might even hinder the debug process!

The Clarion Debugger

The Clarion debugger is the first tool of choice and is activated by enabling Debug mode from the Project properties window. Once a project has been set to debug mode, individual source modules can be selectively enabled and disabled from debugging by right clicking on the source module in the project. This can be useful for simplifying the number of modules presented for debugging when the debugger is activated, but unfortunately each module has to be disabled individually, so it can be a time consuming process.

Russ presented a demonstration application, which was the first app of Devcon to *intentionally* fail.

His debug technique is to run the application in debug mode. When the debugger windows appear create a break point in the module where the error becomes evident. If the source code is extensive it might be necessary to select two break points on either side of the bug, and possibly a third somewhere in between. Press the Go button to run your program until the break point causes the debug windows to gain focus. Take note of any global or local variables that might give clues as to the cause of the bug. Use the "E" button to step over code to the next break points, or until the bug presents itself. Review any changes to variables for more clues.

If the break points indicate that you are near to the bug, use the "T" button in debugger to execute and review the variables for each line of code. If necessary the values or variables can be changed in the debugger, while the code is executing, to see what effect this has on the bug.

If you inadvertently close an exe while the debugger is running it can be restarted from the debugger menu. This is more useful than restarting the exe and debugger from the application generator because the module being debugged is still active in the debugger window.

Russ then demonstrated the use of the Clarion debugger as the Windows default debugger, something I have been reluctant to do. But Russ showed one use of this that has tempted me to reconsider! When a Clarion app produces a GPF it is possible to right-click on the stack trace, and depending on the nature of the GPF you might be able to go straight to the last line of code in your application before the GPF! For those cases where this is not possible, Russ showed the use of the `ASSERT` statement. This syntax:

```
? ASSERT
```

conditionally tests values that it would not be possible to test using `STOP()` and `MESSAGE()` statements.

DebugView

Russ then demonstrated the use of DebugView, a free utility available for [download](#), with vastly superior debug capabilities compared to `STOP()` or `MESSAGE()` statements in Clarion programs. Russ showed how easy it is to implement DebugView, with the help of a free Clarion class and template written by Skip Williams, called "[Debugger](#)" (sic) available from [Clarion Magazine](#).

These tools allow the output of the values of variables, or text, or both, from virtually anywhere you can embed code in your Clarion program, to a window in DebugView, which runs independently of your Clarion program. So instead of popping up messages and pausing your Clarion program, you can document and monitor its execution in DebugView. Anyone who has ever put a `STOP(Fil:Value)` statement within a Clarion LOOP statement should

take note and download these tools now!

Clarion Driver Trace

Russ concluded his presentation with a demonstration and discussion of the Clarion Trace.Exe program shipped with Clarion. Trace can be used with both ISAM and SQL data. It should only be used with one application running at a time, and be sure to shut down the trace utility by un-checking the programs parameters after a debug session, otherwise it will continue to run indefinitely and can greatly impact on performance since all data related transactions are logged to a file!

The Trace utility can also be invoked and disabled by embedding Property statements within your code. This has the advantage of selectively logging data related information, which can also be done conditionally. Russ also suggested using the "?" syntax to enable these statements in debug mode only. To my way of thinking this makes Trace a powerful tool indeed.

In Conclusion

Russ reiterated the need for every programmer to debug programs at some point or other, and by using a bit of logic and the tools outlined he estimated that it should usually be possible for a bug to be identified in fewer than 10 minutes.

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)

online sales and delivery
for your applications & tools

Developer **PLUS**

DevCon 2004: The PHP Templates

by Paul Konyk

Published 2004-09-30

Because of a monitor "malfunction" Pierre Tremblay's session started 45 minutes late. Pierre started off by listing all of the open source software you need to get before you can work with the Clarion/PHP Templates.:

1. PHP from www.php.net
2. ADOdb lib from www.sourceforge.net/projects/adodb/

In addition, some manual editing of the PHP configuration file, php.ini, is needed to change some of the default values so that MSSQL can be used as an ADOdb data source.

Pierre then told us that the PHP language is very popular and quite easy to understand. Next, he explained how he prepared the ADO Northwind app for the PHP templates. First, the Global PHP extension was added and then the language was selected. Pierre (who lives in Quebec) suggested "English" as a reasonable choice. He also mentioned that this was his first DevCon presentation and that he "was not nervous at all."

Throughout the presentation Pierre stressed several points:

1. Use Generate All from the Project menu instead of Make or Run. Since PHP and HTML is the only "code" which is generated you need a browser to test your code. No .exe is ever generated.

2. Two folders are created in the same folder as your app file – "php" and "html". The contents of both of these folders need to be copied to the web server's html base folder.
3. If you already use the Clarion/ADO Templates, you are familiar with these templates.

After showing us the phpdemo app, which used only template code, Pierre customized some display fields and showed us an entry form. He did have a problem with an image which did not display correctly on the update form. Before the next session had finished, Pierre had corrected the bug and showed us his working PHP app. That's pretty impressive – and hour from bug report to a bug fix!

A question and answer session followed with Robert Zaubere fielding some the questions. The estimated price will be \$300 - \$350. There will be a very attractive cross-grade from the ASP templates for about \$100 - \$150. Also, SoftVelocity is planning on producing online training which is similar to the current ADO offering. Finally, the Clarion/PHP Templates should be available in three to four weeks.

[Hyrum Tatton](#) began programming in 1969, while working on masters degree in Chemistry. His computer language experience includes Assembly Language, Basic, Fortran, COBOL, RPG, Neat-3, and Clarion since the DOS days. He now works exclusively in Clarion and COBOL. His professional experience includes time as a pilot, a high school chemistry teacher, university instructor, high school principal, and software developer as CEO of [TES-Software](#). Hyrum really doesn't know what he'll do when he grows up, but he's sure it will be exciting. He is passionate about his family, his church, classical music and golf.

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine



DevCon Pre-Conference Training: Day 3

by Steven Lewis

Published 2004-09-30

While day 1 and 2 of the training sessions focused on new functionality provided by C6, day 3 focused on taking advantage of capabilities available to us since Clarion 4 introduced Object Oriented Programming (OOP) and the Application Building Classes (ABC). The main instructor for the day 3 sessions was Gus Creces, author of the 3rd-party [Handy Tools](#) classes and templates. Much of Gus's tool offerings are based on and derive from the ABC classes built in to Clarion, so he is certainly well qualified to explain how to take advantage of and extend the ABC paradigm. The training materials provided detailed examples and explanations of how to create ABC compliant classes, as well as how to build various types of templates that use these classes.

The main points I took away from the session included:

- Even if you don't write your own classes and templates, understanding how ABC works will allow you to better utilize the existing classes. Given this understanding, you will be more able to locate the embeds you are looking for by examining the ABC class source code.
- Some code and declarations need to be included in the class include and source files to make the classes ABC compliant. This material is pretty simple and much of it can be understood by examining any of the ABC classes and by reading material included in the Clarion 6 on-line help.

- A template wrapper makes it easy to use and derive an ABC-compliant class.
- Writing an ABC template wrapper is fairly simple since most of the behavior comes from adding a few statements within the template. An easy way to accomplish this is to take the boilerplate examples that were included in the training materials.
- The training templates included an EmbedInformation template which assists you in specifying the embed point entries within your templates. Once you have located your desired embed points, the EmbedInformation will generate the proper template code to identify that spot. The ABCFree template set includes a similar template.

Using what I learned in the class, along with some of the provided examples, I have managed to build a template wrapper for an ABC compliant class I had previously written. The class provides the simple capability to export the contents of a list box to a CSV file. The class, template, and a demo application are included with the article. Hopefully this simple example may provide the basis for other template novices to start using these powerful capabilities to extend our productivity with Clarion.

Late in the afternoon after Gus concluded his presentation, Diego Borojovich of SoftVelocity briefly discussed some of the changes that were made in the `FileManager`, `RelationManager` and `ErrorClass` classes for Clarion 6.1. `FileManager` and `RelationManager` are now threaded. This modification was made to restore performance when dealing with large dictionaries and was driven by the new Clarion 6 threading model. Global error handling is now non-threaded with a threaded "helper" class. If you have some code or classes which directly interact with the `ErrorClass`, you may need to rework that code as a result of the `ErrorClass` redesign.

[Download the source](#)

Reader Comments

[Add a comment](#)

Copyright © 1999-2004 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

[Clarion Magazine](#)

Reborn Free

CLARION
online

Clarion News

[Search the news archive](#)

[UK Fenix Training](#)

Radventure and Richard Rose are organizing some Fenix training in the UK on 25 and 26th October at ings Langley. The cost will be €80 (approx £462) which includes buffet lunch both days. Those interested in attending should contact Richard Rose at 01279 469100 or by email.

Posted Monday, September 27, 2004

[Clarion Developers Challenge Announcement](#)

Just a quick reminder for everyone competing in the Clarion Developers Challenge Football Contest to remember to place your picks for this week. Dean has had eye surgery this week and will take a little to get his eyes back focusing correctly, so he'll make an announcement as soon as possible on the weekly prize and Clarion Developers Challenge Vendor of the Week.

Posted Saturday, September 25, 2004

[BoxSoft Super Stuff Status](#)

BoxSoft has almost completed the new Super Stuff docs and examples. In the meantime, you can download the current docs and example program; Super Stuff will be finished before the end of September. For all purchase and upgrade issues (including passwords), please contact Mitten Software. Their e-mail address is Mitten@MittenSoftware.com, and their phone numbers are (800) 825-5461 and (952) 745-4941.

[Clarion Magazine](#)



Clarion News

[Search the news archive](#)

[HotFix 9026](#)

There is a HotFix release to update Clarion 6.1 Build 9025.

Posted Tuesday, August 31, 2004

[HotFix 9027](#)

There is a HotFix release to update Clarion 6.1 Build 9026.

Posted Tuesday, August 31, 2004

[IMDD HotFix 9027](#)

There is a HotFix release to update IMDD Build 9025 to ver 9027.

Posted Tuesday, August 31, 2004

[VTC Template Update](#)

The VTC Version Build template has been updated. This template is free - look under Products|Free.

Posted Tuesday, August 31, 2004

[ClarionMag AutoLogin Utility](#)

Ben Brady has created a utility to allow automatic login to Clarion Magazine. You must be a valid Clarion Magazine subscriber to use it.

Posted Tuesday, August 31, 2004

[Database Migration Tool](#)

Robert Paresi has made available a free tool to help developers migrate between databases.

Posted Tuesday, August 31, 2004

[More Free Clarion Wallpaper](#)

Another free Clarion wallpaper is now available from Jesus Moreno.

Posted Tuesday, August 31, 2004

[CPCS + SV Target Output Enhancement](#)

CPCS has been enhanced to allow for direct output to any specified SV Target Output type, without needing to display a preview or print the report. Please see the "Question and Answers" section of the CPCS help file for details.

Posted Tuesday, August 31, 2004

[xAccess Manager C6.1 Compatible](#)

SealSoft's xAccessManager Lite is compatible now with Clarion 6.1 (Build 9025).

Posted Tuesday, August 31, 2004

[CapeSoft An Official DevCon 2004 Sponsor](#)

CapeSoft is proud to announce that it is an official Sponsor of SoftVelocity's DevCon 2004 in Orlando Florida, USA. CapeSoft will have 3 programmers at DevCon 2004 - Gary James, Jono Woodhouse and Bruce Johnson. The DevCon will run from September 20-22, with pre-conference training running from September 16-18. At the end-of-day Monday (September 20) CapeSoft will be presenting a special "Sponsor's Session" showing off some of the latest CapeSoft accessories. More a technical demonstration than a marketing demo, this will have something fun, something new, something old and more than a few Wows. Plus there will be no less than 15 CapeSoft product give-aways at the DevCon.

Posted Tuesday, August 31, 2004

[MAV Direct ODBC Update](#)

MAV Direct ODBC new versions (009 and 007) have been released. Library version 0.09 fixes a problem with NVARCHAR fields and improves internal library and class optimization, and is still \$99 during beta. Template version 0.07 adds multi-column SQL single tree browse, new SQL autonumbering mechanism, QBE, and extended

drop list features. The template version is also \$99 during the beta.

Posted Tuesday, August 31, 2004

EasyExcel 3.06

Ingasoftplus has released EasyExcel 3.06, with bug fixes, some changed methods, a number of new methods and new templates. This version is for Clarion 5.5 and 6.1 (9026). Free for all registered customers. The password has changed; to get the new one contact Ingasoftplus and provide the Order number and Registered user name.

Posted Tuesday, August 31, 2004

EasyMultiTag 2.07

EasyMultiTag 2.07 is now available. This release adds ClarioNet support and bug fixes. Fully compatible with the Clarion 5, 5.5 and 6.1.

Posted Tuesday, August 31, 2004

EasyListPrint 1.09

Ingasoftplus has released EasyListPrint 1.09. Changes include the ability to enforce Excel format for a column, and several bug fixes.

Posted Tuesday, August 31, 2004

Ripley Code Commentor Updated

The Ripley Code Commentor has been updated with a number of new features, including auto-detection of product updates.

Posted Tuesday, August 31, 2004

XP-Taskpanel 1.5

Version 1.5 of the PowerOffice XP-Taskpanel is now available. Many new features have been added, and bugs have been fixed.

Posted Tuesday, August 31, 2004

PDF-XChange & Tools Version 3 Now Gold And C6.1.9025 Compliant

PDF-XChange and Tools SDK's Version 3 are now GOLD and fully C5 through Clarion 6.1.9025 compatible. As far as possible compatibility

with the major third party report templates has also been achieved (as well as ClarionNet from C55 onwards). PDF-Tools SDK upgrade is \$249.00, PDF-XChange 'Pro' SDK upgrade is \$399.00.

Posted Tuesday, August 31, 2004

BST 3.62 Patch

A bugfix is available for BST. This fixes a problem with the monthly booking template loading records from the following month under some circumstances. The main installer and demo have been updated as well. Install the patch, reload your app, compile, that's it. The IDE will pick up the changes.

Posted Tuesday, August 31, 2004

Fomin Report Builder 2.90

Registered users are welcome to download the latest update to Fomin Report Builder.

Posted Tuesday, August 31, 2004

CPCS Builds For C6.1 9026

All CPCS products have been recompiled with Clarion 6.1 #9026. There were two bugs fixed in this build: Incorrect previewer display under Win98/ME (this was a C6.1 RTL bug which SV fixed in 9026); "Incorrect embed for AutoBind" error, corrected in the CPCS templates when a filter or range was specified via prompts.

Posted Tuesday, August 31, 2004

EasyAutoEntry 1.04 For C6.1 (9026)

EasyAutoEntry (Full and Demo versions) 1.04 have been recompiled with Clarion 6.1 (build 9026).

Posted Tuesday, August 31, 2004

EasyMultiTag 2.06

EasyMultiTag 2.06 has been recompiled with Clarion 6.1 (build 9026).

Posted Tuesday, August 31, 2004

New Gitano Software Builds for C6.1 Build 9026

New installs are available for C6.1 Build 9026 for all Gitano Software

utilities.

Posted Tuesday, August 31, 2004

ProDomus 6.1 Updates

ProDomus has the following available for Clarion 6.1 build hotfix 9026. These all work with Legacy and ABC applications. PD Browse Button Lookup 61-02 . All Source Code Browse Button Class Library with Auto-Complete, multiple lookups without aliasing, optional required matches, SQL support, and more. PD Translator Plus 61-02. Multi-language translation class libraries and utilities. This includes auto-picture translation and conversion of ENTRY controls to TEXT controls at run-time in support of double-byte entry and a source code parsing utility for extracting all source strings. PD 1-Touch Date Tools 61-02. Date entry handling for all windows in one step: the addition of the global extension to your application. Typically, each date entry has advanced spin control features, a calendar button, and a multi-featured popup calendar. All you have to do is populate an entry or spin control with a date picture and the templates and library do the rest. Range limits can be set in the dictionary or embed code. Includes multi-language international functionality such as picking up day and month names and abbreviated names from the users windows registry without any further coding. Week of the year calculations also reflect user registry entries. There is no upgrade charge from Clarion 6.0 to 6.1.

Posted Tuesday, August 31, 2004

Ingasoftplus Builds For C6.1 (Build 9026)

New installs are available for C6.1 (Build 9026) for: EasyDocker; EasyHelper; EasyReport; Easy3DStyle; EasyListPrint; EasyResizeAndSplit; EasyExcel; EasyAnimation; EasyMultiTag.

Posted Tuesday, August 31, 2004

New solid.software Web Site

Jens reports that he has finally had the time to update his web site. If you have a minute, please visit.

Posted Tuesday, August 31, 2004

Clarion Domains For Sale

Jesus Moreno has some Clarion domain names for sale, including ClarionJobs.com, ClarionProfessional.com, ClarionExpert.com, and ClarionForum.com.

Posted Tuesday, August 31, 2004

Firebird Lessons 50% Off

Robert Meyer is offering 50% off of the "How to use Clarion and Firebird 1.5 together" lesson set. A sample of a lesson is available. Lessons 9 is about complete, and Robert has several more in the works. Once he publishes the 15th lesson, the promotional is over.

Posted Tuesday, August 31, 2004

BoxSoft Super Invoice 6.16 and Super Tagging 6.16

BoxSoft has updated New Super Tagging and SuperInvoice with bug fixes, as well as some new features. You can download all the Super Templates from www.boxsoft.net. For all purchase and upgrade issues (including passwords), please contact our distributor, Mitten Software. Their e-mail address is Mitten@MittenSoftware.com, and their phone numbers are (800) 825-5461 and (952) 745-4941.

Posted Tuesday, August 31, 2004

vuFileTools 2.0

vuFileTools Version 2.0 now includes 72 functions to help you manage files and the Windows environment. There are functions that let you Copy, Move, and Delete groups of files and folders, Create shortcuts, Hide the Desktop and Start Bar, Dynamically resize the screen, the easiest system registry functions you'll find anywhere, and a whole lot more! Please visit the web site and view the Quick Reference Guide for a complete list of these functions.

Posted Tuesday, August 31, 2004

Sterling Data And C6.1

All Sterling Data templates are 6.1 compatible

Posted Tuesday, August 31, 2004

Copyright © 1999-2003 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Posted Saturday, September 25, 2004

vuFileTools Demo

A demo version of vuFileTools is now available. This version includes the manual, quick reference, and both compiled demo programs. These EXEs demonstrate all of the features of vuFileTools version 2.0. Fly-Over help is enabled on each button of the demo apps to help explain each function. These demo apps are fully functional, and the only difference between the demo DLL and the commercial DLL is a splash screen when the DLL is first loaded and unloaded.

Posted Saturday, September 25, 2004

Solace Software Releases New Live Update Templates

Solace Software has released the first Beta of their new Live Update templates. This is a new template and class which are added to your application so that your program automatically checks for latest updates of itself, downloads the files and updates the exe together with any other files you need to add to their system. This gives your application a more professional approach as well as removing problems associated with sending customers updates via email or CD. To activate an update, you simply copy your updated program files to an FTP server and the templates will automatically detect latest versions. If you do not have an FTP server available, then you can rent space on the Solace Software server.

Posted Saturday, September 25, 2004

ADDA 1.0.4

ADDA (Advanced Data Dictionary Architect) version 1.0.4 is released.

Posted Saturday, September 25, 2004

RUpdate 1.b Beta Released

Rupdate version 1.b is now available. This version allows you to specify the name of the application in the web update screen. This version also now works from Clarion for Windows version 2.03 up till Clarion version 6.1 Current users, please note that the password has changed, so email bdl@riebens.co.za for the new password. RUpdate is a Clarion for Windows Template that allows a developer to implement the following measures into Applications created with the

ABC and/or Legacy template chains within minutes: Provide a method for the end user to check the Web for application version status; Provide for downloading the new version file(s); Provide for running the version update file(s).

Posted Saturday, September 25, 2004

Clarion Developers Challenge Week 2 Winner

First place for Week 2 of the Clarion Developers Challenge goes to.....Jason Miner! Jason picked a very solid 11 out of 16 to claim first place in this week's contest. Timo Multanen and Bob Gove finished a close second with 10 correct picks. For finishing first in this week's Clarion Developers Challenge, Jason will be receiving a copy of vuFileTools donated by Bill Roe and Valutilities.

Posted Saturday, September 25, 2004

ThinkData 30% Off Sale

ThinkData announces a 30% off sale on all products on the ThinkData web site from September 1 to- October 15 2004, in celebration of DevCon. All of products are Clarion 5.5 and Clarion 6 and Clarion 6.1 compatible, contain both ABC and Legacy templates, and ship with complete source code. xmlFUSE, regularly \$249, now \$174; qbFUSE, regularly \$199, now \$139; OutlookFUSE, OutlookFUSE, regularly \$199, now \$139; EnhancedScrollClass, regularly \$69, now \$48.

Posted Saturday, September 25, 2004

Clarion Developers Challenge Week 1 Winner Announced

First place for Week 1 of the Clarion Developers Challenge goes to.....Ken Bame! Ken picked a very solid 13 out of 16 to claim first place in this week's contest! Jim Henry, Al Lucas and Joe Henry all finished a close second with 11 correct picks. For finishing first in this week's Clarion Developers Challenge, Ken will be receiving a copy of the Ripley Code Commenter donated by Scott Ripley and Ripley Software. Thanks, Scott and Ripley Software for your generous support to the Clarion Developers Challenge and the Clarion community!

Posted Friday, September 17, 2004

CapeSoft Support Notice

Due to DevCon2004, and other events, many of the CapeSoft technicians will be out of the office between the 13th and 30th of September. A lot of support emails will probably go unanswered during this period until their return. Thank you for your patience.

Posted Friday, September 17, 2004

SelfService Goes Gold

CapeSoft SelfService allows your Clarion 6 application(s) to be run as a Service in Windows NT, Windows 2000, Windows XP or Windows Server 2003. SelfService has completed it's Beta phase, and is now officially in Gold status. Of course, as usual, this also means the special price of \$59 is ending. As of Friday the 24th of September, the price will be \$79.

Posted Friday, September 17, 2004

New HotDates From CapeSoft

Date and time-based information, such as birthdays, leave, etc is often easier to read when organized on a calendar than when displayed as a list. Patterns can be more easily spotted. (For example - is that guy always sick on a Monday?). How about a scheduler/planner to view a summary of a days events and edit it from a popup? Enter HotDates. Use HotDates to view dates in a calendar - drill down to month, or day. Create a scheduler. Print your calendar straight to a printer. Create a customizable date picker. HotDates supports Clarion 5.5 and up, currently ABC templates only, including Multi-DLL. HotDates is currently on special during Beta testing for \$199; gold price will be \$299.

Posted Friday, September 17, 2004

New GPFReporter From CapeSoft

GPFReporter allows you to trap program "illegal operations" and offers the user the opportunity to email a short report to the developer detailing the illegal operation. GPFReporter identifies the line number in your application source code where the illegal operation occurred, and reveals the chain of procedure calls which preceded the illegal operation. GPFReporter requires HyperActive and

WinEvent. The only down side is that it only works on Windows XP and Windows Server 2003. Programs containing GPFReporter will however run normally on other versions of Windows. GPFReporter supports Clarion 5 and up, Legacy and ABC, including multi-DLL. GPFReporter is currently on special during Beta testing for \$49; gold price will be \$79.

Posted Friday, September 17, 2004

[New AnyFont From CapeSoft](#)

AnyFont allows runtime changes to your default program font. You can change the font name, size, colour, style and character set. All this can be stored in an INI file for the next time the program is run. Give your users the ability to customize their applications look and feel. AnyFont also automatically adjusts the size of the program's windows to compensate for the change in font parameters. AnyFont supports Clarion 5.5 and up, Legacy and ABC, including multi-DLL. AnyFont is currently on special during beta testing for \$29; Gold price will be \$39.

Posted Friday, September 17, 2004

[Free Sybase ASE Express Edition For Linux](#)

A free developer version of Sybase Adaptive Server Enterprise Express Edition for Linux is now available. Also available for Linux: Sybase Enterprise Application Server and Sybase SQL Anywhere Studio.

Posted Friday, September 17, 2004

[ClarionMag DevCon Sale](#)

Clarion Magazine is celebrating DevCon 2004 with its biggest sale of the year! New subscriptions (and renewals past the grace period) are now just \$99 (save \$26). Remember, subscriptions and renewals now include all the back issues. Russ Eggen's Programming Objects in Clarion is just \$74.95 (save \$15) The Tips & Techniques book is just 59.95 (save \$30!) The SQL & Databases book is just 59.95 (save \$30!) And that's not all! With each purchase you get this amazing set of Ginsu steak kni...nah, just kidding. But if you are a current subscriber, or you are buying a subscription or renewal, you'll also

receive an additional \$10 off both the Tips and SQL books! This discount will be applied to your order at the checkout.

Posted Friday, September 10, 2004

DeveloperPLUS Store Temporarily Offline

The DeveloperPLUS store will be offline for a few days to facilitate software upgrades and modifications, and should be available again on September 13.

Posted Friday, September 10, 2004

Clarion Developers Challenge Vendor of the Week

The Clarion Developers Challenge Vendor of the Week for Week 1 is...Well, he's 41 years old, a husband and father of two teenage girls. For two years he's been running his own software contracting business out of his home. He has 17 years programming experience, with a little over 11 of those being in Clarion (yes, all the way back to Clarion for DOS). He enjoys Star Trek (all generations and movies) and Stargate SG-1. Just like most of us he enjoys twiddling with computers but cringes when he has hardware problems. Give up? The Clarion Developers Challenge Vendor of the Week for Week 1 is...Scott Ripley and Ripley Software! Scott will be donating to the winner of this Week's Clarion Developers Challenge a copy of the [Ripley Code Commentor](#). The Ripley Code Commentor lets you quickly and easily add in-line comments to your code when making code additions, changes or deletions.

Posted Friday, September 10, 2004

PowerOffice DevCon Specials

PowerOffice is having a September DevCon special with big discounts on all products.

Posted Friday, September 10, 2004

OutlookBar 1.2

Version 1.2 of OutlookBar is now available for download. Due to PowerOffice products appearing on a warez-site the installation passwords have been changed. To obtain the new password(s), please send an e-mail including your full name, and e-mail address

used when purchasing the product(s) (the date of purchase would also be helpful) to clarion@poweroffice.no.

Posted Friday, September 10, 2004

Clarion Developers Challenge Now Open

The Clarion Developers [football] Challenge is now open! All Clarion Developers or employees of Clarion shops are eligible to compete. (Yes, that means even the secretary, and even vendors that offer prizes.) Prize lists are still being formed, but there will be having weekly and Grand Prize winners. Rules are posted on the web site and Dean Burgess or the other Clarion Developers who have competed before will be glad to answer any of your questions. The pool password is: ABCTemplates

Posted Friday, September 10, 2004

Updated CMag Automatic Logon Utility

Ben Brady has posted a new update for the Clarion Magazine Automatic Login utility. You can now maximize the window in order to read Clarion Magazine more efficiently.

Posted Friday, September 10, 2004

Clarion Third Party Profile Exchange Updated

The Clarion Third Party Profile Exchange consists primarily of profiles of third party add-on products and vendors. This includes freeware templates and tools as well. Online and Downloadable Profiles available. Online product profiles include Product Internet URL, Order URL, Currency code, Dated Price Quote, Grouped by Category, Clarion 6.x Compatible, Extended Description and Download Page Reference. Currently, there are 481 product profiles and 501 vendor profiles. You must have Product Scope 32 PRO Version 5.0 to view profiles with data files (downloadable profiles). 304 Clarion 6.x compatible products as of this release.

Posted Friday, September 10, 2004

New gReg Site

Jesus Moreno has redesigned the gReg site to make information easier to find. The template used on the web site is one of Jesus's

Ready-to-go templates, available from [1stLogoDesign](#).

Posted Friday, September 10, 2004

[Comparison Between the Two Approaches to Achieve Net-Centricity](#)

A little, um, light reading from Mike Gorman (and Bruce Haberkamp) on the tag and post method versus the data standardization approach.

Posted Wednesday, September 08, 2004

Copyright © 1999-2003 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.