



Clarion magazine

Volume 1 Number 9 - October, 1999

Issue Index

- Main Page
- Log In
- Subscribe
- Open Source
- Links
- Mailing Lists
- Advertising
- Submissions
- Contact Us
- Site Index
- ClarionMag FAQ
- Download PDFs
- Search ClarionMag

DeveloperPlus
Your source for
development tools
and add-ons.
Your outlet for
application sales.

[DevCon Details: Welcome And Keynote Address](#)

In his keynote address, Hank Asher, founder of eData, outlined the future of application service providers and offered a unique service to Clarion developers.

Posted on October 6, 1999

[Seen And Heard At DevCon](#)

More pictures and MPEG video from the '99 DevCon!

Posted on October 6, 1999

[DevCon Details: Web Edition 2 and iBuild@TopSpeed](#)

Details of Bob Zaubere's general session covering the upcoming Web Edition 2 and iBuild@TopSpeed products.

Posted on October 6, 1999

[DevCon '99 Special Subscription Offer Extended To October 15th!](#)

The DevCon '99 special offer for new subscribers (20 months of Clarion Magazine for \$99, a regular value of \$125) has been extended until October 15th. For \$99 you get all the back issues, all the October issues, and a year's subscription beginning November 1.

Posted on October 14, 1999

[Guest Editorial: Andrew Guidroz II](#)

Is it a Sun-ny day? Andrew Guidroz II muses on Linux, Solaris, Java, and the future of Microsoft.

Posted on October 14, 1999

[DevCon Details: Richard Chapman On TopSpeed's Future Plans](#)

In his Wednesday session at DevCon Richard Chapman, TopSpeed's VP of R&D, elaborated on iBuild, C6, and the future Java direction, and gave some insight into the workings of the London development centre.

Posted on October 14, 1999

[A Project Development Methodology](#)

Free download: TopSpeed's project development methodology document provides helpful information on managing projects and the client/consultant relationship. Essential reading for independent developers. Many thanks to Rich Mebane for making this document available.

Posted on October 14, 1999

[Tool Talk: New Products At DevCon](#)

Tom Hebenstreit, Clarion Magazine's Reviews Editor, goes wild at the DevCon '99 third party exhibits.

Posted on October 14, 1999

[Open Source Update: SQL Class](#)

Sebastian Talamoni has contributed an SQL class and template wrapper to simplify the use of embedded SQL.

Posted on October 14, 1999

[Editorial: DevCon Wrapup](#)

Dave Harms rolls up his editorial shirtsleeves and reviews TopSpeed's future direction as revealed through DevCon '99.
Posted on October 20, 1999

[Andrew's Kitchen](#)

Andrew Guidroz II explains what TV dinners have to do with Clarion embeds.

Posted on October 20, 1999

[Open Source Update](#)

Chris Behling has updated his all-Clarion graphing example app with some bug fixes.

Posted on October 20, 1999

[ConVic '99 Clarion Conference](#)

If you're in Oz, or will be Nov 5-9, be sure to take in ConVic '99, the Victoria Clarion developers' conference. Last year's event was extremely well-run, and this one looks to be a repeat. Speakers include Arnor Baldvinsson, Andrew McPherson, David Blundell, Simon Brewer, Owen Brunker, Des Cousens, Bruce Cowan, and Yogi Loechner.

Posted on October 20, 1999

[David Bayliss On The RelationManager - Part 2](#)

David Bayliss concludes his two-parter on the RelationManager class.

Posted on October 26, 1999

[The Novice's Corner: Understanding Clarion Code](#)

The Novice's Corner series continues (after a summer/DevCon hiatus) with an introduction to the Clarion programming language.

Posted on October 26, 1999

[The Clarion Advisor: Detecting Duplicate Records](#)

Gordon Smith presents a utility class for detecting and fixing duplicate records.

Posted on October 26, 1999

[October 1999 News](#)

Clarion world news: product announcements, upcoming events, and more.

Posted on October 26, 1999

[Product Review: Clarion Source Search](#)

One of the hottest third party products at DevCon was Carl Barnes' Clarion Source Search. Tom Hebenstreit reviews the must-have CSS version 1.5.

Posted on October 26, 1999

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

DevCon '99 Special Report

DevCon Details: Welcome And Keynote Address

by Dave Harms

Roy Rafalco, [TopSpeed](#)'s President and CEO, started off DevCon '99 with an introduction of TopSpeed staff, old and new. Among those mentioned were Richard Chapman, VP of Product Development, Ole Poulson, VP of Technology, Frank Watts, Senior VP of Sales and Marketing (who comes to TopSpeed from [Logic Works](#), publisher of ERWin), Esli Badenhorst, VP of Marketing (who formerly worked with Frank Watts at Logic Works), and Rich Mebane, VP of Technical Services (formerly VP of Global Services for [Newbridge Networks](#)). Rafalco also introduced Stamos Fafalios, one of TopSpeed's directors (and a Clarion user) and had the international distributors and Team TopSpeed (TTS) stand. He noted that this was the largest TTS turnout of any DevCon.

Rafalco then made a number of announcements, including [eData](#)'s sponsorship of a happy hour after the day's session and a tour of their [Boca Raton facility](#) on Wednesday.

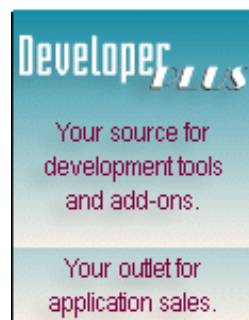
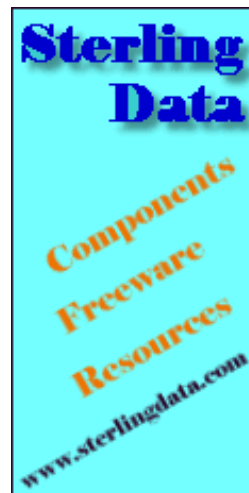
Next came a summary of TopSpeed's current situation. The company has experienced record revenue growth for the last several years. The new marketing team has been hard at work, and there has been a 25% increase in the number of new users over the first 8 months of 1999.

Rafalco did have an internet-related warning for the attendees. He referred to the massive capitalization of internet-related firms, the shift of business to the internet, and the fundamental changes coming as business completely retools to take advantage of the new technologies. He referred to this change as a revolution, not an evolution. He quoted the [Wall Street Journal](#) on the fundamental shift away from PCs to the web, and also on Microsoft seeing itself as at risk if they don't change their model. [The Gartner Group](#), the consulting firm TopSpeed uses, is predicting that next year the number of programmers targeting Windows will drop from 60% to 40%, a "staggering thought" for Rafalco. "People aren't asking what Windows development tools they will use, they're asking what web development tools they will use."

According to [Forrester Research](#) Java/CORBA programmers outnumber Microsoft COM programmers two to one. TopSpeed has felt this shift as well, and now sells twice as many Web editions as Enterprise editions.

Rafalco pointed out that the worst answer is to ignore the changes and do nothing. He said that the need to rewrite for the web is a great opportunity, which leads to the theme of the conference, iDevelop@TopSpeed. Rafalco then introduced Bruce Barrington (the creator of the Clarion language and the founder of Clarion Software, which together with JPI became TopSpeed Corporation).

Barrington's job was to introduce the keynote speaker, and began by referring to last year's DevCon and [Citrix](#)'s Ed Iacobucci, Ernst & Young's 1998 Entrepreneur of the Year. Barrington said he'd landed an even



bigger fish for this year's conference: [eData's](#) Hank Asher. Barrington first met Asher some 11 years ago when Asher and Roy Brubaker (who now works for TopSpeed) did work on multiprocessing servers. Asher was already a Clarion developer at that time.

Asher later moved to Florida and set up [Database Technologies Online](#), which is now a NYSE company with a market capitalization of about \$500 million. DBT provides data to investigative and law enforcement organizations. In 1998 Asher began eData.com. eData is getting a lot of attention, including two visits from [Sun Microsystems'](#) Scott McNealy in the last six weeks. eData has the investment banking community whipped "into a complete and utter frenzy," according to Barrington, who then introduced his friend and colleague and "the guru of application services," Hank Asher.

Asher began by recollecting his days at earlier DevCons, and how he looked up to the other developers. He also corrected an earlier comment by Barrington that he (Asher) had left Indiana because his efforts were undercapitalized. In fact, said Asher, he was fired on the grounds that he couldn't get along with people and his ideas didn't work. "I think I have proven that my ideas do work," he said, and paused significantly. When the laughter died down he offered the opinion that the opportunities are unbelievable for those who capitalize on what's happening.

Asher related his first experience with Clarion, its high productivity, and its imposing creator ("Bruce Barrington! Sounds like a character out of a romance novel! 'Bruce Barrington entered the ballroom and the ladies stopped.'")

Comments on Bruce's imposing persona and a few other jokes aside, Asher pointed out the great opportunities for application service providing, and that as a hayseed from Indiana he knew enough to make hay when the sun shines. [Click here for an MPEG \(611k\)](#). He then commented on the [Java](#) implementations he'd been told were coming down the line. (This wasn't elaborated on until Wednesday's presentation by Richard Chapman.) Asher did say that both Scott McNealy and Bruce Barrington are saying the same things about the computer industry.

According to Asher billions of dollars are now available in venture capital, but the era of very small independents striking it rich is over. "If we can band together" and offer solutions the return would increase enormously, said Asher, adding that the opportunities for developers collaborating are immense. Asher's formula for success is passion for product, and being in the right place at the right time.

At eData it wasn't easy to convince the investors to go with TopSpeed development products for any of the work. When the in-house developers couldn't get some internal data maintenance applications done, Tom Moseley and Richard Taylor came on board and had quick success, also doing some web development with Clarion. All of eData's products are now driven by TopSpeed.

DBT has a data engine (written in C) which is "extremely powerful." Asher said that the first time he ran a report on himself it found both his parents, all his siblings, his in-laws, his ex-in-laws, his ex, "her newest victim," all his old addresses, and his old neighbors and where they were now. He asked himself "Oh my God, what have we done?"

When Asher ran a report on his new system at eData, he had the same reaction, only instead of a 2-3 minute response it was a 4-5 second response.

Asher then offered to everyone in the room a data engine and a database of billions of public and commercially-available records which can be integrated with a developer's application. This is the system that prompted Sun's Scott McNealy to say "I don't think you know what you've got!"

Following this Asher made a pitch for TopSpeed's partnership support program. He suggested taking the partnership program to the next level. This would facilitate developers sharing products, tools and technologies, and making the most of the opportunities that are there. He proposed forming a TopSpeed partnership alliance, and addressed concerns over how such a program would be administered by referring to the scarcity mentality versus the abundance mentality. Asher's feeling is there's more than enough for everyone, and "we'd have to be real greedy and real stupid not to make this thing work."

Asher was then assisted by Tom Moseley who ran the eData demonstration which involved searching the database using a TopSpeed front-end. Asher pointed out the file system allows for some advanced data retrieval techniques, the amount of data going into the database will be huge, and the TopSpeed (presumably this means Clarion) data access tool will be able to utilize the database better than any other. Asher stated he would like to "offer [the data access tool] to the TopSpeed community."

Queries assembled by the client software are sent to eData, the report is created, NCOA (change of address) processed, packaged, and sent back. The software also handles label printing and other tasks.

Asher concluded the demonstration with the comment that "these are serious times for serious people to take serious advantage of, and this hayseed believes it's very, very sunny."

When asked what opportunity there would be for Clarion developers to work with eData's software and database, Asher indicated he would like to open that up through the TopSpeed partnership program. Elaborating on the contents of eData's 30 terabyte database, he stated it contains information on "every individual in the country" including their past locations, which allows for trend analysis. As well there is a definitive list of dead people, and sources report that there is extensive financial information available as well.

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

In This Issue

[DevCon Details: Welcome And Keynote Address](#)

Posted on October 6, 1999

[Seen And Heard At DevCon](#)

Posted on October 6, 1999

[DevCon Details: Web Edition 2 and](#)

[iBuild@TopSpeed](#)

Posted on October 6, 1999



Clarion magazine

DevCon '99 Special Report

Seen And Heard At DevCon

By Dave Harms

Photos by Dave Harms, Larry Teames, and Russ Eggen

The last DevCon of the millennium (okay, so that's being just a little grandiose, not to say technically inaccurate) has wrapped up, the cleaning crew has removed the last discarded third party flyers and washed the drool out of the carpets, and life in Fort Lauderdale and in Pompano Beach is returning to normal, whatever that is. Here's one more look at the DevCon that was.



Richard Chapman, TopSpeed's elegantly dressed VP of R&D, keeps a sharp eye out for the paparazzi.

The week kicked off with registration - here Arnold Young gets his license to cause trouble from Susie Buchler.



Chubby Bunny
MPEGs
At the Key West
Fest Arthur
Barrington hosted a
contest in which
the entrants had to
stuff as many
marshmallows as
possible in their
mouths and still be
able to say
"Chubby Bunny"

[MPEG of AR
explaining the
contest](#) (368k)

[MPEG of the
contest in progress](#)
(304k)

[MPEG of the winner](#)
(320k)



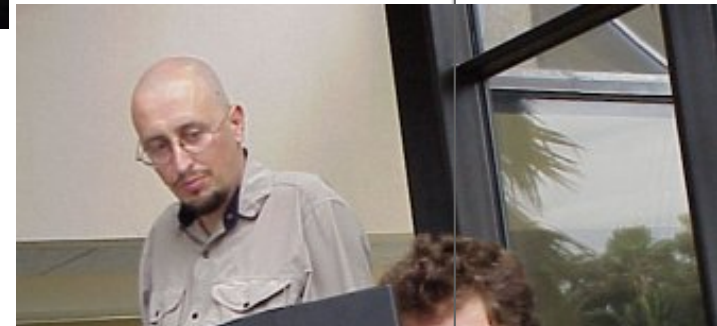
The Hyatt Regency is located next to a marina (in fact there is a yacht brokerage as well, so if you're so inclined you can purchase the yacht of your choosing, say, after the last conference session and before happy hour).



It may be a bit difficult to make out the lettering on this yacht, but if you look closely you'll understand why Larry Teames has been one of the last third party vendors to embrace ABC.



From left, Brian Staff, Andrew "Cajun" Guidroz II, Bernie Groperrin, Mike Pasley, and the back of Bruce Johnson's head. Obviously these guys don't have enough to do.





RKC tries to use the debugger as the least-recognized man at DevCon (aka SuRF) looks on.

It's David Bayliss! Ah, no, the hat fooled me. It's just Andy "Cowboy" Stapleton.



Arnor Baldvinsson (left) arrived from Iceland, on his way to Australia. Craig Ransom anxiously tries to get the cameraman's attention to explain why there appears to be a plant growing out of Arnor's head.



Pete Halsted (NextAge Consulting) travels with his Sony Vaio system, rather than a regular laptop. Note the flat panel display and the slimline desktop. The whole thing fits neatly into a carryon bag. The picture below also shows the Vaio mini-laptop.



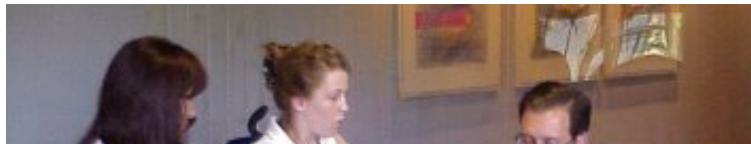


Here's Bob Z (Zaunere) and Mini-Z, who always gets nervous when Bob starts waving his hands around like that.

Is this thing pointed the right direction? Your roving reporter tries to figure out his new digital camera.



If you could read that name tag, you'd find it says Alexey Solovjev. Alexey is something of a legend in Clarion circles. Before he went to work for TopSpeed he used to submit bug reports in which he not only described the problem, he described where in the original source code the fix should be made. He determined this by disassembling the object code. TopSpeed wisely decided it was best to have Alexey on their team.





Susie Buchler, Kelly Zaunere, and Esli Badenhorst compare notes.

Lunches were served in the ballroom, a short walk from the hotel and adjacent to the marina. This was also the site of the Key West Fest.



In a moment of sudden terror, Cowboy wonders if Scott Ferrett was right.

Carl "Do you need a Clarion search program?" Barnes ropes and hog-ties another customer. Even the hotel staff are looking over his brochures.





For those who couldn't handle the sessions anymore, there was always life at the pool.

Ah, well, the less said about this the better, eh Brian?



The London developers were treated to a cruise aboard Hank Asher's yacht, which reportedly costs \$71 per minute to run. What does that come to per developer per hour?



Lisa Simmons, former TS developer and now reported to be the project leader at eData, enjoying the yachting life.



From left, clockwise, Stamos Fafalios, Bruce Barrington, Hank Asher, Ole Poulson, Kelly Zaubere, Scott Ferrett, and Michael Ledet (eData).

These guys are all wearing white shirts. Must be the crew, which makes you wonder why they aren't busy swabbing the decks. From left, Gavin Halliday, Richard Chapman, Andy Ireland, and Bu...Tom Moseley (who works for eData).





What would DevCon be without a secret meeting or two? Here some of the TTS guys hang out at an undisclosed location with a top TS official whose identity must be protected. From left, Andy Stapleton, Brian Staff, Dave Harms, Ross Santos, Mr. X, Bruce Wells, Mike Pickus, and Andrew Guidroz II.



Steve Parker's been contending that the Internet Connect support team are identical twins (yes, I know there are four of them). He's right; they look so much alike, I can't tell them apart. In alphabetical order: Arnor Baldvinsson, Jim Defabia, Steve Parker and Mike Pasley.

Here's the new, improved Bruce Wells!
Well maybe just a bit too improved at the moment.





And finally, the last word in cool: Tommy Tequila and Charli Carlson.



Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

In This Issue

[DevCon Details:
Welcome And
Keynote Address](#)

Posted on October
6, 1999

[Seen And Heard
At DevCon](#)

Posted on October
6, 1999

[DevCon Details:
Web Edition 2
and
iBuild@TopSpeed](#)

Posted on October
6, 1999



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

DevCon '99 Special Report

DevCon Details: Web Edition 2 and iBuild@TopSpeed

by Dave Harms

On the second day of DevCon '99 Bob Zaunere gave an overview of the new features in Web Edition 2, with a bunch of Team TopSpeed members doing "the wave" in the back of the room as he took the podium.

Zaunere began with a brief summary of Clarion history, beginning with the Clarion 3GL language and CPD model files through to the current product and the trend to applications being web-enabled. He asked for a show of hands of those who have Web Edition, and a large number indicated they did.

Performance issues with the existing Web Edition have been addressed with Web Edition 2. Java applets are no longer required, as all controls are displayed as straight HTML. This allows webmasters to make changes to the appearance of the app without needing to understand Clarion coding. WE2 also uses POST instead of GET which fixes some problems such as using large memo fields. There is also improved cookie support.

New in WE2 is a simple scripting language called TSScript.

Testing is now easier as there is an option to link the app broker into an EXE. Doing a make invokes the browser so that local testing looks the same as testing across the internet. The ABC templates and classes have also been updated (Joe van Niekerk covered a number of new features in his track session, including classes for Winsock/email and new language features like INTERFACES).

Jim Defabia took over briefly to demonstrate a WE2 version of an application he'd created for a previous day's session on the current WE product. That first application, an employee status board showing who is in or out of the office, used Java classes to reproduce on the client machine a list box and various controls. The WE2 version used all HTML controls to reproduce the same functionality.

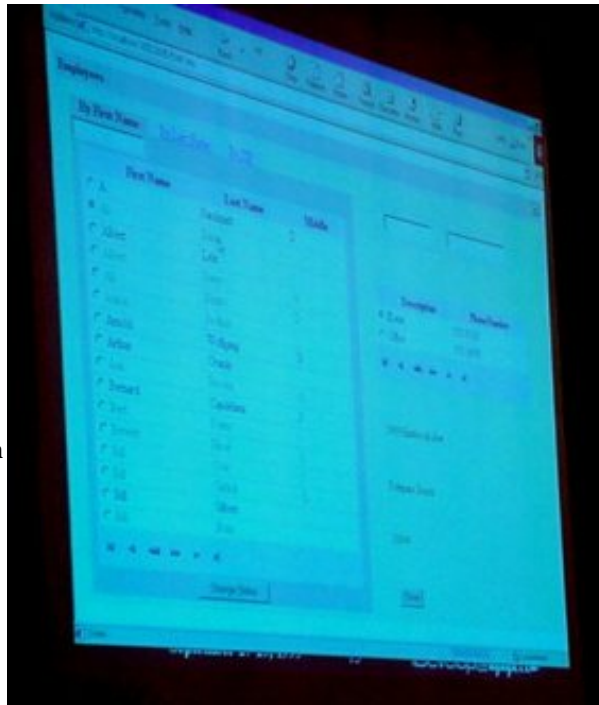
Sterling Data
Components
Freeware
Resources
www.sterlingdata.com

Developer Plus
Your source for
development tools
and add-ons.
Your outlet for
application sales.

[Click Here for MPEG video](#) of Bob Zaunere explaining what's new in Web Edition 2. (1384k)

As the image shows (if not very clearly) the list box is an HTML list or table with navigation buttons rather than scroll bars. There is a child list box and a status update button which brings up a change form. There is also an entry locator at the top of the list. A radio button indicates which record is selected.

With the Java WE application a change to the list box data results in a change to the list box only. The page itself is not resent. With the WE2 Java-free application any change to the list box results in a new page being sent, but as the amount of data is very small this happens quite quickly.



One advantage of using a non-Java approach is data can be copied to the clipboard, since everything is straight HTML. (At present there is no DHTML generated by WE2.)

Defabia demonstrated changing the look and feel by substituting a different set of HTML templates which the application used at runtime. In WE, which generates the HTML at runtime, this change would have required a recompile.

You can still have a dual-mode Windows application if you wish.

No upgrade pricing has yet been set. Zaunere expected a beta to be released sometime in October. He pointed out the key features were the increase in speed and the integration with existing web sites.

iBuild And AppKits

Zaunere went on to talk about the new iBuild@TopSpeed product, which is positioned to make the creation of web-enabled applications easier. TopSpeed's strategy includes creating horizontal application development kits which developers can modify. TopSpeed will also provide application hosting via an arrangement with eData (which has a massive fiber infrastructure in place, according to other reports).

There are three app kits planned for the initial release: iSell@TopSpeed (e-commerce), iMarket (contact management), and iPublish (content management). Also included in the iBuild product is a full release of [Dreamweaver](#) (expected to increase the cost by under \$100). There is a new version of the app broker which can run applications on secondary workstations on a round-robin basis. This approach can also be used for redundancy.

App kits include Clarion templates, HTML templates, DLLs if needed, dictionary, and wizatron style sheets.

iSell comes with storefront and store manager components, including a wide range of features. Initial transaction support will be for [CyberCash](#).

The development cycle, according to Zaubere, involves generating the application which creates the initial web pages, then switching to Dreamweaver and making changes. These changes are merged back into the application by means of wizards when you task switch back to the application.

There will be an option to generate static HTML pages for use by search engines.

iSell planned features include category and product management, special pricing, product options, inventory management, customer account management, security, store manager security, taxation, order tracking, and non-CyberCash transactions.

iPublish, the content management app, lets users manage web site content directly by filling out forms. This could involve new HTML documents (such as press releases), products, menus, etc.

iMarket is an email marketing/contact management app kit. When kidded about iSpam@TopSpeed, Zaubere replied (tongue planted firmly in cheek) that this would be in a subsequent release.

These app kits can all be integrated to work within a single web site.

In This Issue

[DevCon Details: Welcome And Keynote Address](#)

Posted on October 6, 1999

[Seen And Heard At DevCon](#)

Posted on October 6, 1999

[DevCon Details: Web Edition 2 and iBuild@TopSpeed](#)

Posted on October 6, 1999

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Guest Editorial

Andrew Guidroz II

I'm just sitting around thinking.

So much of my work in the computer business is spotting future trends for my company and customers. Who will be the next big player in the operating system business? What languages will be dominate? Which companies will direct the technology advances of the next 20 years in the computer business?

Microsoft has dominated the desktop environment for some time now. But we're beginning to hear rumblings from other players in the market. Linux is beginning to gain mindshare and marketshare although the majority of the activity is on the smaller server range. Sun is rapidly gaining mindshare and marketshare in two distinct areas. One is the high end server market. The other is in the use of the Java language.

Now, on the language side, it is obvious that even Microsoft is hearing the threat from Java. Recently, top managers at Microsoft have publicly talked about how they've forgotten the development community and how they need to refocus on that area. They've also made large investments in Inprise, the former Borland. Why would they do this? Isn't Delphi a competitor to Visual Basic? I believe it is to keep major players who provide tools which are dependent on the Windows API in business. The Windows API dependency is the key to Microsoft's domination of the industry.

Sun's Java language promises platform independence: there are no Windows API strings attached. But this also promises Sun platform independence. You are no longer tied to using their Solaris as your foundation to use their technologies. On the operating system side, Solaris still is one of the biggest players for huge shops. NT is there also but in larger numbers, smaller applications. Solaris seems to be used as a single machine solution whereas NT is used as a "machine for each application" solution.

Where is Linux in this equation?

Linux sits on the fringes. It moves into a shop to do some small task at first, like e-mail or web hosting. But, gradually, it does perform larger tasks. Most shops that I see it in have multiple Linux boxes, just as they have multiple NT boxes, each doing a separate task.

Linux also is the "open source" operating system. This means that the source code used is open for all to see and you are even welcome to enhance it for your own use. This part of the picture has the other players nervous.

Your source for
development tools
and add-ons.

Your outlet for
application sales.

- Translator PLUS
- Date Tools
- Address Formats
- Lookups
- Finance

International Software

Microsoft's reaction has been to announce that they are considering releasing portions of their operating systems to the development community, although nothing has been seen yet. Microsoft has also funded benchmark testing of NT vs. Linux and even posted a page describing the "Myths of Linux". They surely see Linux as a direct competitor.

Sun has announced a modified open source plan to release some of the Solaris source code. It isn't nearly as far reaching as the one Linux falls under but is a step in that direction.

I can understand Linux vs. Microsoft and I can understand Sun vs. Microsoft. But what does this really mean about Linux and Sun? Nearly everyone I know who is doing things with Linux isn't touching Java at all. They are using PHP and CGI and all those wonderful server side things that were designed by operating system geek for operating system geek. And they like using C++ because they like to get down to the metal of the machine.

Java creeps in because of you and I. We are programmers and we want the robustness and insulation of a real programming language. We've worked at the client side and want to leverage some of the same power we have had there onto the server side. Does this mean that Sun and Java and Solaris are rivals to Linux? Well, Java does run on Linux so, although the majority of Linux folks aren't using it, it doesn't mean that new programmers on the platform are not going to use it. Solaris is positioned toward the high end and Linux toward the lower end.

But I feel there are intrinsic differences between Sun and the Linux community that make them very nearly competitors. Torvalds, the man who began the Linux revolution, made light of Sun's latest open source efforts just a few days ago. And certainly Java's platform independence implies independence from all operating systems, including Linux.

Who wins? Who has the solution? I'm not certain it is an either/or solution. I have many different servers running in my shops with applications written in many different languages. This is as fragmented as I have ever seen the computer industry. That, to me, is the key point.

The industry will probably never go back to having a single dominant company. Microsoft may or may not still reap the lion's share of the earnings but other companies will have a tremendous amount of input as to the direction of computing.

Andrew Guidroz II is an active member of the TopSpeed community on the Internet and writes software for the insurance/finance industry as well as a host of other industries. His famous Cajun cookouts have become a central feature of Clarion conferences throughout the US. Andrew's Cajun website is <http://www.coonass.com>.

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

In This Issue

[DevCon '99
Special
Subscription
Offer Extended
To October 15th!](#)

Posted on October
14, 1999

[Guest Editorial:
Andrew Guidroz
II](#)

Posted on October
14, 1999

[DevCon Details:
Richard Chapman
On TopSpeed's
Future Plans](#)

Posted on October
14, 1999

[A Project
Development
Methodology](#)

Posted on October
14, 1999

[Tool Talk: New
Products At
DevCon](#)

Posted on October
14, 1999

[Open Source
Update: SQL
Class](#)

Posted on October
14, 1999



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

DevCon '99 Special Report

DevCon Details: Richard Chapman On TopSpeed's Future Technology Direction

by Dave Harms

Wednesday afternoon Richard Chapman (RKC) gave a presentation on TopSpeed's future directions, including the forthcoming Web Edition 2 and the iBuild product, and touched briefly on TopSpeed's Java direction. He also provided some information on the members of the London development center.

As in Bob Zaunere's presentation, a bunch of Team TopSpeed guys at the back of the room punctuated the speaker's introduction by doing "the wave."

RKC promised to be very vague about product release dates. Right now the London team is working on the Web Edition 2 (WE2) upgrade and iBuild@TopSpeed, two closely-related products.

The Internet Connect 2 technology which is in WE2 cannot be used with C5B or any earlier version – you only get it with WE2. WE2 will also have some IDE enhancements, mainly involving Wizatrons (which have been enhanced for the iBuild technology). There is also a property inspector for the IDE (think of the Wizatron property inspector). RKC stressed that the upcoming WE2 beta will contain a lot of stable, proven code: the HTML generation area is new but the rest of the system had to maintain a lot of compatibility with C5B. London expects that the beta will be useful for everyday work. The message was buy the C5 WE upgrade today and get access to the beta.

iBuild is the other major focus at the moment. There are some major differences between iBuild and WE2, such as the timing of HTML generation. WE2 generates HTML pretty much the way WE does (at runtime). This provides the greatest flexibility for rearranging controls and so forth at runtime but the downside is increased server load and more complexity in controlling the application's appearance (although performance has been improved in WE2). In most cases the controls on the screen don't change; only the data displayed by those controls changes. In iBuild the HTML is created when the application is generated rather than at runtime, and just the values of the use variables are changed at runtime.

Generating HTML sooner in the cycle gives the web designer a chance to work with the page contents without having to compile the program.

TopSpeed is also working on the three [AppKits](#) planned for iBuild. The creation of these AppKits by the Development Centre involves adding ABC support where needed (i.e. Winsock, email, invoice generation, audit trails), creating templates, and adding wizatron support. Finally a wizatron style folio is created for the AppKit.

One other difference between the products is iBuild has a 32 bit IDE, and WE2 has the 16 bit IDE.



Clarion 6 Web Edition And Java

RKC pointed out that Clarion 6 Web Edition is Clarion 6; only one product is planned. His advice for current C5 user is to upgrade to Web Edition ([click here for MPEG video](#) - 367k). C6 WE can also product desktop apps, of course.

Offering information on C6, RKC warned that what he was about to present was subject to change.

C6 will be the best of iBuild and WE2. The most significant enhancement is that the entire IDE will be 32 bit (much applause). This also allows the use of some of the more modern OS features, such as slider and RTF controls (which could not be used previously because of the need to maintain 16 bit compatibility). The announcement that 16 bit targets would not be supported was also greeted with applause.

C6 will also include the early HTML generation technology from iBuild.

Technologically, there will not be a lot of difference between iBuild and C6. The products are aimed at different markets. iBuild is intended for web developers who have HTML skills but not database skills. C6 opens up the iBuild feature set with full embed points and team development capabilities.

RKC noted he'd been asked if he was going to mention the "J-word." "Yes," he said, "I am," and paused. After laughter from the audience, he continued, saying that Clarion is an unbeatable product for desktop and client/server applications. But what about distributed, multi-tier scalable web applications? "Distributed programs are hard to write. It's a different paradigm, a different mindset, a different set of problems. To address this, we need to re-engineer the whole runtime side [of the Clarion language]...We could update the Clarion language with all the features we needed to support this re-engineering. In the process we'd have to change it so you wouldn't recognize it. So we had a better idea. Let's let Sun do our compiler maintenance."

RKC described Sun as having the language (Java) already suited to the task, with features it would be hard and expensive to put into Clarion, such as multi-platform support. Also Java is an industry standard, which means when a new technology comes out the providers make sure it's accessible from Java. This relieves TopSpeed from having to come up with support for that technology.

"Java is a very nice language," said RKC, "[and] many of the concepts are similar to Clarion, particularly if you're working in the ABC paradigm." TopSpeed's mission is to hide the complexity of creating distributed applications, with which developers currently struggle.

The London Development Centre

Although the TopSpeed head office is in Pompano Beach, Florida, development is coordinated from London, England. RKC gave an overview of the development team members, many of whom telecommute, some from as far away as St. Petersburg and South Africa. He stated that TopSpeed's recent recruitment drive covered the entire world.

Note: Many of the development team members, particularly team leaders, have extensive math and/or computer science education from prestigious universities (i.e. Cambridge, Oxford), while others originally came to the attention of the development team as customers and sometimes as third party vendors.

The following summary focuses on the past and present work of the seventeen Development Centre members.

Richard Chapman, VP of Research and Development

- Joined JPI in 1990, head of the Development Centre since 1994



- Developed the Clarion runtime library

Ole Poulsen, VP of Technology

- Borland's third employee, and co-founder of JPI
- Originated the Internet Connect concept
- Main focus is the application broker



Nigel Hicks, Director, Product Development

- Co-founder of JPI
- Developed Clarion IDE framework, Report Writer

David Bayliss, Director, Product Development

- Joined JPI in 1990
- Designed ABC, Wizatrns, compiler, leads Application Paradigms teams

Gavin Halliday, Director, Product Development

- Studied physics at Oxford
- Developed AppGen, leads IDE enhancements team
- Working on Web Edition 2



George Barwood

- Co-founder of JPI
- Designed/developed the TPS driver
- Working in David's team on a Java file system (FileClass)

Claudia Hwang

- Worked on SQL drivers, Dictionary Synchronizer
- Working on AppKits and ABC features.

Roy Hawkes

- Working in David's team on Wizatrns, templates, and ABC features (does most of the template work)
- Formerly with tech support

Jake Smith

- Developed the IC Java classes
- Working in David's team on Wizatrns (and the engine that lays out controls) and AppKits



Andy Ireland

- Formerly with TS Consulting
- Working in David's team on Wizatrns and ABC features, including the Winsock classes



Joe van Niekerk

- Based in South Africa
- Working in David's team on ABC features



Dr. Mourad Mechkour

- Worked on a document clustering system for web information retrieval
- Working in David's team on a Java file system (FileClass)

Ilka Ciocanel

- Worked on Dictionary Editor and Dictionary Synchronizer
- Working in Gavin's team on IDE features, the 32 bit port of the IDE, and the iBuild product

Gordon Smith

In This Issue

[DevCon '99](#)
[Special](#)
[Subscription](#)
[Offer Extended](#)
[To October 15th!](#)
Posted on October

- The user interface guru (and author of Compile Manager and Clarion Class Library browser)
- Worked on TopScan and new IDE features
- Working in Gavin's team on IDE features and HTML templates for WE2
- Based in Ireland

Scott Ferrett

- Joined JPI in 1991
- Developed the SQL file drivers
- Working on Nigel's team on driver enhancements



Alexey Solovyev

- Worked on low-level runtime library features
- Working in Nigel's team on low-level runtime library support
- Based in Russia



James Cowan

- Worked on network bandwidth monitoring/throttling
- Working on a Java distributed application model

At the end of the session RKC took questions. He was asked about the status of the new OLE/COM layer. The completion of this work required changes not suitable for an interim release, so it's not clear whether it will be in WE2, but RKC will be recommending that it is (to loud applause). He also said that C6 and Java are "two separate items in our future, not one and the same."

14, 1999

[Guest Editorial: Andrew Guidroz II](#)

Posted on October 14, 1999

[DevCon Details: Richard Chapman On TopSpeed's Future Plans](#)

Posted on October 14, 1999

[A Project Development Methodology](#)

Posted on October 14, 1999

[Tool Talk: New Products At DevCon](#)

Posted on October 14, 1999

[Open Source Update: SQL Class](#)

Posted on October 14, 1999

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Embedded Secure Document

The file <http://www.clarionmag.com/v1n9/pub/services.pdf> is a secure document that has been embedded in this document. Double click the pushpin to view services.pdf.





Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

DevCon '99 Special Report

Tool Talk: DevCon 99 Special Report

By Tom Hebenstreit, Reviews Editor

Welcome to Tool Talk, a periodic column on happenings in the world of Clarion products, templates and tools.

With over a dozen third party tool vendors demonstrating and selling their wares, this year's DevCon provided a great opportunity for attendees to both experience products firsthand and to meet the people behind the products. DevCon also provides the perfect time for vendors to announce new products, and that's the topic for this column: new and updated tools announced or demonstrated at DevCon.

Note: Not all of these products are actually available yet, and some may change a bit before release. Please follow the links or email the vendors for the most up-to-date information.

Cowboy@YourElbow
from [Cowboy Computing Solutions](#) and [TS Resources](#)

SQL guru Andy "Cowboy" Stapleton and George Willbanks of TS Resources are teaming up to create a new series of interactive training CDs aimed squarely at the Clarion SQL/Client-Server market. Called "Cowboy@YourElbow," the series will eventually be comprised of seven CDs (or more) with each CD covering one area of expertise. Topics will include Beginning SQL, Intermediate SQL, Advanced SQL, SQL Tools (i.e., using Data Modeller or PowerDesigner/S-Designer), Replication and more.

Cowboy and George were showing the first in the series (Beginning SQL), with various tasks being visually demonstrated step by step (click by click, if you prefer) while Cowboy's voiceover described what was being done and, more importantly, why it was being done. The user can go directly to various segments via an index, and segments can be paused or restarted as needed. If you started at the beginning and worked through the entire series, it looks like you would acquire a fairly complete education on using SQL with Clarion. As a bonus, you would also get a good working knowledge of how to speak Texan (thanks to Cowboy's inimitable presentation style). Seriously though, this is a very interesting new product and I'd like to see more training tools like it.

The cost is expected to be US\$95 per CD plus shipping and handling. Please be aware that they are in early production right now; you cannot purchase the set as of this writing. A website is also in the works (www.yourelbow.com - what else?), but it is not up yet. Keep an eye on the Clarion Magazine news section for announcements regarding this product.

Clarion Source Search
from [Carl T. Barnes](#)

After one of the longest betas in third party history (Carl was showing a



CSS beta last DevCon), Clarion Source Search was finally officially released just prior to DevCon. What is it? What it sounds like – it searches Clarion source files (any text files, actually) for whatever string you specify and then shows that string in context. This is only one of the features that differentiate CSS from standard search programs; it understands Clarion source files.

Need to find where a variable is used? Where it is declared? CSS will find it and display not only the line of code where it was found, but the procedure, routine or method name where it was used as well. Double clicking on a match opens up a view of the source file. Trying to understand ABC? Want to find the base code for a particular ABC class method? A couple of clicks and bingo, there you are.

A ton of options let you save and restore search specs, skip comments, find labels, filter/save/print result sets, use wildcards and more. CSS also supports using regular expressions, an extremely powerful way of creating what are in essence string search formulas. Think of them as super wildcards on steroids, and you'll be getting close. There's even a Regular Expression Assistant to help write and use these search formulas.

CSS lists for US\$45.99. More information and a demo can be found at <http://www.carlbarnes.com>

I highly recommend checking it out; it's hard to understand just how useful CSS is until you've actually used it. Watch for a full review in an upcoming issue.

PD Translator Plus from [ProDomus](#)

Phil Will, the king of translation tools, released an enhanced edition of his flagship PD Translator templates, with the new product being designated PD Translator Plus. Unlike the previous version which worked with C4/5 Legacy/ABC (and which will still be available), Translator Plus has been totally OOPed and tightly integrated into C5 and the ABC classes.

Phil ran me through a quick demo of Translator Plus, and it seems to be both powerful and complete. A wide variety of templates, classes and utility programs appear to cover all the bases, making it as easy as it can be to internationalize your applications. To quote from the brochure:

"Scalable class and utility components for building multi-lingual applications. Picture Class auto-translates pictures using international information for over 90 locales. String and Type classes consolidate translations into a single file. Environment and Character Classes set the Clarion Environment and handle extended characters. Utilities extract strings from source code, create translation files, create international environment files, and assist in translation."

Whew! Lots of stuff there. Look for the full review of PD Translator Plus in an upcoming issue of Clarion Magazine. Also, Phil mentioned that he has updates of his PD Date and Lookup tools coming down the road.

The nine components that comprise PD Translator Plus can be purchased separately or in a variety of pre-packaged combinations. Prices for the packages range from US \$199 for the Basic Edition to US \$875 for the Enterprise Edition.

For more information on PD Translator Plus or the entire line of ProDomus products, visit: <http://www.prodomus.com>

TimeSavers Scheduler from POSitive Software Company

After years of people asking for some type of easy to implement scheduling/calendar solution, we now suddenly find ourselves with not one but two products to choose from (see next item as well). The first set to actually hit the streets are John Hickey's Scheduler templates, released as part of his TimeSavers series. These templates were hot off

the press, so to speak, having been released the Friday before DevCon started. Two major templates are included: a Calendar Creator template for creating calendars and a Scheduling Grid template that lets you create Monthly, Weekly or Daily Scheduling Grids. Additional features include drag and drop, icon support, a sample app demonstrating how to create schedulers and more.

The templates support C4 and C5, Legacy and ABC.

TimeSavers Scheduler is available now, and is being offered at an introductory price of US \$99 through the month of October. More information and a demo of the templates can be found at:

<http://www.clarioncentral.com>

Watch for a complete review in an upcoming issue.

G-Cal Pro
from [Gitano Software](#)

Jesus Moreno of Gitano Software told me that his long awaited G-Cal Pro scheduling and calendar product is nearing completion. Features on the calendar side include six calendar types, a wide variety of date functions, a date calculator and an event announcer that lets you dynamically reschedule events or run other programs when an alarm goes off.

On the scheduling front, over a half dozen views and variations ranging from daily to monthly are provided. Appointment features include multiple categories, multiple event types (user defined), recurring events, colors, icons, alarms, priority levels, To Do lists, drag and drop, appointment queries, a number of built-in reports, hooks to Report Writer and more.

G-Cal Pro is being offered for a special price of US \$149 which is good from now through the first week after the product's official release. Owners of the current G-Cal product can also get an additional discount.

More information can be found at: <http://www.gitanosoftware.com>

Hopefully, we'll be seeing this one soon. And yes, the magazine will have a full review once it is released.

Data Modeller
from [PeaBrain Software](#)

Data Modeller holds a rather unique position in that it is both a third party tool and an integral part of TopSpeed's Clarion 5 Enterprise Edition. Roelf DuPreez, the author, was showing what looks to be a major upgrade/expansion of the existing product.

Here is a list of new features, as put forth by PeaBrain in their flyer:

- A complete new Entity data design mode. Draw hierarchical diagrams of your application or database design before you start to physically create the files.
- A new Logical view allows you to see the logic of your design or subset at a glance.
- A totally new Subset module lets you create unlimited subsets.
- A new Wizatron view allows you to see the results of the TopSpeed Dictionary Wizatron or how the Wizards will make use of your control properties.
- A new Query builder allows you to query data from your dictionary across multiple files. This allows you to plan views or find incorrect data easily.
- New SQL script creation options allow you to easily convert existing or new dictionaries to an SQL database script.
- A new Attachment editor allows you to store additional documentation and/or SQL Stored Procedures, Update Procedures and Triggers in the dictionary.

What can I say? I want it now!

Pricing for the update (or whether there will be an update charge at all) was not set at the time I talked to Roelf. He anticipated a release date somewhere in the range of a few weeks after DevCon, i.e., early to mid October. Keep an eye on the PeaBrain web site for the latest news on this increasingly useful (indeed, essential) tool:

<http://www.peabrain.co.za>

[DeveloperPlus](#) and [ClarionShop](#)

Hmmm. Seems like things are happening in pairs lately.

With the TopSpeed Accessories Program undergoing major changes (a topic to be covered in a future column), two separate parties have seen (and jumped on) an opportunity to provide secure one-stop online shopping for Clarion third party tools.

The first site, [ClarionShop](#), is being created jointly by [CapeSoft Development](#) and [NextAge Computing](#). It looks to be focusing mainly on third party tools, and is aiming to provide additional content such as reviews (hey – competition for me!), user product ratings, opinions and more in addition to online shopping. A preliminary version of the site is already up, and you can get a preview at: <http://www.clarionshop.com>

[DeveloperPlus](#), on the other hand, appears to be focusing more on the publishing and product fulfillment end of things. The initial flyer announcing the site indicates that they will be offering a wide range of services to third party tool and utility developers. Even more intriguing, they want to provide those same services to you for your own vertical market applications. These are some very interesting possibilities opening up here. In any case, see for yourself: DeveloperPlus has just announced the grand opening of their site at:

<http://www.developerplus.com>. Oh, yeah, in case you were curious, Lee White of [Lodestar Software](#) is the driving force behind DeveloperPlus.

I'll be covering both sites more in depth in the future as they get people signed up and their identities and directions become more firmly established.

Incidentally, both of these sites allow the vendor to keep a larger chunk of the selling price of their tools than the original TS Accessories program. Given that this is not exactly a huge market to begin with, I heartily approve of anything that gives the vendors more incentive to keep improving existing products as well as encouraging the creation of new products.

Wrapping Up

My overall impression after DevCon is that the Clarion Third Party scene is not only active, it is expanding. And that makes me very happy since, as a certified tool-head, I rely on these people to constantly make me more productive. Hey, "The more tools the merrier" is my motto. (You should see the size of my Registry file!)

New tools, new vendors and new directions from TopSpeed look to make the next year an exciting one in the Clarion world. I can't wait!

Comments? Suggestions? Write me at: reviews@clarionmag.com

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

In This Issue

[DevCon '99 Special Subscription Offer Extended To October 15th!](#)

Posted on October 14, 1999

[Guest Editorial: Andrew Guidroz II](#)

Posted on October 14, 1999

[DevCon Details: Richard Chapman On TopSpeed's Future Plans](#)

Posted on October 14, 1999

[A Project Development Methodology](#)

Posted on October 14, 1999

[Tool Talk: New Products At DevCon](#)

Posted on October 14, 1999

[Open Source Update: SQL Class](#)

Posted on October 14, 1999



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

The Clarion Open Source Project

Product Index

Updated October 20, 1999

The following products are available from Clarion Magazine under the [Developers Open Source Public License \(DOSPL\)](#).

- [Debugging/Profiling](#) - These classes from CoveComm Inc. generate trace logs and application execution logs. Template included. Version 1.1.
Posted Aug 3, 1999
- [BitList Management](#) - Class and template from JS Software (Jeff Slarve) to bitslice LONGs and display checkboxes for setting the individual bits. Version 0.9912. Very slick.
Posted August 5, 1999
- [File Utilities](#) - A set of classes by Patrick O'Brien to handle file dialogs and splitting file names. Includes an example application.
Posted August 7, 1999
- [Debug Classes](#) - Richard Rogers' class and template for debugging/tracing.
Posted August 9, 1999
- **UPDATED!** [Graphing Classes](#) - Chris Behling's example application does graphing using all-Clarion code. Latest update includes some bug fixes.
Updated October 20, 1999
- [Trackbar/Slider](#) - Pierre Tremblay's 32-bit slider class and template as described in his Clarion Magazine [article](#).
Posted August 19, 1999
- [Graphing Learning Example](#) - James Cooke's learning example does OOP graphing using all embed points.
Posted September 5, 1999
- [Wizatron Style Viewer](#) - a database application by Sebastian Talamoni for viewing/managing style sheets. 386k, includes EXE
Posted September 20, 1999
- **NEW!** [SQL Class](#) - a class (and template) from Sebastian Talamoni that simplifies the use of embedded SQL.
Posted October 14, 1999

If you've placed some code under the DOSPL and you'd like to have it listed here, email cosp@clarionmag.com

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



In This Issue

[Editorial: DevCon Wrapup](#)

Posted on October 20, 1999

[Andrew's Kitchen](#)

Posted on October 20, 1999

[Open Source Update](#)

Posted on October 20, 1999

[ConVic '99](#)

[Clarion Conference](#)

Posted on October 20, 1999



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Editorial

DevCon '99 In Retrospect

by Dave Harms

If I had to choose three words to sum up this year's Florida DevCon they would be, in order of emphasis, marketing, Internet, and Java. Note that I said "order of emphasis" and not "order of importance," at least not as far as many developers in attendance were concerned. There was in fact a certain amount of grumbling over the number of sessions devoted to marketing, some of which explained TopSpeed's direction, and others which offered direction to Clarion developers in their own efforts.

Marketing has always been a weakness at TopSpeed, so the new marketing team members deserve full points for their recent success in getting new customers (most of which, by the way, are buying the Web Edition). And there's no doubt that many of us need more marketing smarts in our own businesses. But for what is usually a technical conference, the amount of non-technical information was a bit jarring.

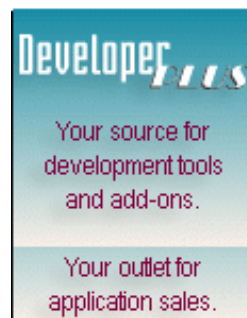
Some of this information could be explained as gap-filling. In previous years attendees have received beta releases of products. At this year's DevCon no such distribution was made, and while there was some interesting news about upcoming products, clearly TopSpeed didn't feel ready to provide many details.

The first major announcement was about the upcoming Web Edition 2, which offers a "caffeine-free" approach to delivering Clarion applications across the internet. WE2 is a good direction, and important direction for TS. It's probably a little bit behind the market, but because it's a Clarion product it leverages the high productivity of the Clarion environment and so should be a real boon to developers.

The most important aspect of WE2 is that it no longer relies on Java on the client side in order to deliver web applications. All the controls are now HTML controls. This is an absolutely essential step forward, and one that does at least provide the hope that Clarion developers will be able to create and deploy powerful, fast Internet applications with considerable ease. As important as this step forward is, however, this is not as dramatic news as we've become accustomed to getting at DevCon. WE2 is more of a dot release (after all, some independent developers have been doing caffeine-free Internet Connect for some time).

Most of the really interesting news was in the announcement of the longer-term product direction. The first big step for TopSpeed is iBuild. Unlike WE2, which generates HTML at runtime, iBuild works with HTML created at design time. This HTML can be created in a real HTML editor, such as the to-be-bundled DreamWeaver.

Wizatrons connect the HTML and the application so that changes in one are reflected in the other. But iBuild is more of a web developer's tool (relying heavily on wizatrons) than it is an application developer's tool, or that's how it appears. The key is the deployment of wizatrons as the two-way binder between the HTML and the application.



iBuild doesn't get fully opened up until Clarion6 arrives, at which time the iBuild technology and the Clarion technology we now know are merged back into a single product. How successful the temporary product forking is remains to be seen, and it's also not clear to me how much of this split is driven by marketing and how much is driven by the rate at which London makes the new web technology available.

Despite some uncertainty over the iBuild marketing focus, it became apparent at the conference is that TopSpeed is serious about pursuing a new direction. I remember a DevCon some years ago at which Bruce Barrington, talking about the technical features of the upcoming Clarion Database Developer, said "It's not your daddy's Clarion anymore." Well, it's not your daddy's TopSpeed anymore either. The company has previously stated its intent to go public at some point. In order to do this TopSpeed clearly needs a relevant, useful Internet development product.

When I first became a Clarion developer I had a great enthusiasm for the product. I wanted and expected the world to realize what a fantastic tool it was. As time went on and the Clarion community didn't really grow, I became resigned to working with something that not too many people had heard of. I'd see people become Clarion developers with the same enthusiasm that I had once showed. I'd hear them say what a great product Clarion was, and how everyone should know about it, and I'd smile to myself and think that given a little time they'd come around and realize it isn't going to happen.

Well, I think it could happen. TopSpeed could break through. I'm not sure just what the odds are, but for the first time in quite a few years I believe the company really does have the potential to become something much more than a little-known vendor of a little-known product used by a loyal, dedicated band of followers.

TopSpeed knows it needs to leverage its code generation capabilities. That, along with its knowledge of how to build business application frameworks, is what the company does best.

I think it's essential, however, to make a distinction between this code generation/application frameworks capability and the Clarion language itself. I remain skeptical that a proprietary language from a small company is going to gain, on its own, wide acceptance. For that reason I'm very pleased to hear TopSpeed announce that they will, in a future release, generate Java code in addition to Clarion code. The marriage of TopSpeed's code generation capability and application development knowledge with the Java language seems to me the best of all possible worlds. And for once TopSpeed's timing may be just about right.

When Java first came on the scene it was touted for its thin client capabilities. Sun's vision (as you might expect from a company heavily into server deployment) was of software stored in centralized locations and downloaded to client machines as applets. The applet vision hasn't materialized as expected (as the current Web Edition demonstrates), partly because of slow Internet connections, partly because of Java's immaturity as a language, and no doubt for other technical and political reasons as well.

Consequently there was a shift in focus from using Java on the client side to using Java on the server side. Servlets have sprung up in place of applets, and Java Server Pages compete with Active Server Pages. At the same time, Java is gaining ground in the development of distributed, multi-tier applications.

Now, some five years after the introduction of Java applets, Java is starting to hit its stride and it is being seen as the natural successor to C++. It is getting the speed it needs, the APIs, the third party support, and the industry acceptance. We may well be on the verge of Java's breakthrough as the standard language for most kinds of application development. If TopSpeed can deliver a Java code generation tool at the right time, I think they'll do very well as a company.

In This Issue

[Editorial: DevCon Wrapup](#)

Posted on October 20, 1999

[Andrew's Kitchen](#)
Posted on October 20, 1999

[Open Source Update](#)

Posted on October 20, 1999

[ConVic '99](#)

[David Harms](#) is an independent software developer and the co-author with **Ross Santos** of *Developing Clarion for Windows Applications*, published by SAMS (1995). He is also the editor and publisher of Clarion Magazine.

[Clarion Conference](#)
Posted on October 20, 1999

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag



DeveloperPlus

Your source for
development tools
and add-ons.

Your outlet for
application sales.

Why I Hate TV Dinners

Or

Eating TV Dinners Embed?
Don't Get Sauce On The Sheets!

by Andrew Guidroz II

Sometime during the alpha of C4....

- David Bayliss: "With overriding and the new priority settings of embeds, you have, in essence, an infinite number of embed points."
- Larry Teames: "Just not where we want them."

You know, I really hate TV dinners. I mean I despise them. What could be worse?

For those of you who don't know what I mean by TV dinners, let me try to explain. Around 40 years ago, a man named Clarence Birdseye experimented with freezing prepared food. The Swanson company took his research and came up with an individual meal all precooked and frozen. All you had to do was pop it into the oven for about 30 minutes and voilà, instant meal. The name "TV Dinner" was used because the marketers envisioned consumers glued to the front of their television sets and not wanting to spend a lot of time fussing over the stove. This was a meal you could prepare and eat and never miss a minute of television time.

TV Dinners come in a foil container with a foil lid so you never even have to open it or see it until it is done. And that is exactly what makes me hate them so.

You see, you're stuck with whatever is inside. Let's say you bought the great American meal, the turkey dinner. There is a bit of turkey meat with gravy in the first compartment, mashed potatoes with butter in the second, and green peas in the third. It has already been cooked, already seasoned, the choice of side dishes has been made for you. You really

have no control. You may be able to sprinkle a little seasoning on it after you take it out of the oven but that's about it. How bland! How boring!!

What if you wanted to put a little bacon in those potatoes? The flavor will never be the same as it would have been if you actually cooked the bacon with the potatoes. Or maybe you wanted a white sauce and this sauce is brown. And did they stuff the turkey with the seasonings you enjoy so much? And what if you're allergic to peas?

That's the beauty of cooking things yourself. You can alter the combination of foods. You can add just the seasonings you like. And this process doesn't stop at the initial preparation. You can alter the recipe as you are cooking! Not salty enough? Add a little. Notice that the tomato plants in the garden have some ripe tomatoes? Just add them in. You have an infinite number of places that you can alter your cooking plan.

I think that is the reason why a lot of us like Clarion so much. The IDE in conjunction with templates, the wizards, and soon the wizatrions, generate a nice skeleton application. I can then interject my own code in various places to get just the flavor I'm looking for.

But can I really? Can I put my code anywhere? Is that a TV dinner I smell in the oven when the Clarion splash screen pops up?

What Clarion Was

I began using Clarion for Windows when it was first released in version 1.0. That was the beginning of the Legacy templates. The templates had those marvelous embed points scattered throughout. They had nice labels. I was really productive using them.

But times changed. The templates were modified, enhanced, enlarged, and made more complex. And I found that with my apps, I really couldn't place my code exactly where I wanted it. So, I began coding embed points with the OMIT statement. And dropping in IF tests to avoid executing some of the Legacy code. I was cutting and pasting generated code into my own embed points. Then, some of the embed points had very similar names. The embed points weren't documented so the only way I could learn what they did was to stuff some code in there and read the generated code.

Embeds in legacy provided you with an easy way to interject code but were poor in providing you with a mechanism to replace or enhance code.

There were some routines in most procedures but there were no embed points mapped directly to them. In other words, you couldn't tell exactly what routine you were modifying or where by adding embed code.

I could modify the templates and place my own embeds into the system. But there the maintenance nightmare began. Each new version of Clarion would force me to study the templates and repeat all my changes.

A procedure with only a window in it has about 61 embed points and 107 lines of code. That works out to an embed every $1\frac{3}{4}$ lines. When you're designing a splash screen 61 embed points are a lot. But that is a horribly small number when you are writing apps that have hundreds of controls on a single window. There are complex things happening that you really need to get inside of.

I could still do what I wanted to do but my productivity was falling as I wrote apps that were nontrivial.

What Clarion Is

Then along came ABC and the newer IDE. Everything looked so familiar and yet all of the old rules went out the window.

A procedure in ABC with only a window has about 240 embed points and 58 lines of code. That works out to an embed point every $\frac{1}{4}$ line of code. That is a seven fold increase over the legacy templates. How is this possible?

Little Is Routine

Because most of the work is now being done in classes, there is far less source code generated. These classes rely on methods (which are procedures) to do all of the work. There are few, if any, routines generated in your source. So does less code mean fewer opportunities to embed code? Did Topspeed give us less power with ABC? Nope. OOP comes to the rescue.

OOP allows you to override any method. This means that you can put your code in place of the code Topspeed has written and have it executed instead. No more OMIT. No more bracketing code with IF tests. This kind of functionality is built directly into an OOP language and ABC and the IDE take advantage of it. Every time you write embed code, you're really replacing a method that Topspeed has written with your very own. The code is custom and does what you need because you have written it.

But wait – if it replaces Topspeed's code, you don't want to rewrite everything, do you? The early alpha testers complained that putting a single line remark in an embed point caused a procedure not to execute. That remark literally replaced an entire method, and that method may have been an entire browse. But the templates do a really nice thing for you. They automatically insert a bit of code that says:

```
ReturnValue = PARENT.MethodName
```

This calls the original method, the Topspeed code, for you. So you now have the power to write embed code before and after Topspeed's code. All you have to do is place your code either before or after the call to the parent's method. This means two embed points per method, a before and an after. And you can avoid Topspeed's code by placing your code before the parent call and RETURNING before the parent is called.

You'll even notice that the procedures have template generated code that overrides the default behavior of the classes. Topspeed's templates are using the power of overriding and embedding just as you the programmer are.

Associating each embed with a method has a lot of other great side effects also. It means that you have an easy way to relate the embed point back to the generated source. If you can read the method name, you can easily search the generated source for that method and find your embedded code.

You don't have to cut and paste anymore either. You can just type:

```
Object.MethodName
```

And off you go, calling Topspeed's code where ever you feel like it. No more finding out that Topspeed fixed a bug in the generated code and you have to find everywhere you pasted it to fix it manually.

It also means that each embed point is easy to document. The method name can be descriptive but also easy for you to match up in the Application Handbook. The Application Handbook is the one Clarion manual I read every single day. It is rich in information that teaches you how methods relate to each other, how different ABC objects relate, and it has code examples that show you how the methods are used. It is available in PDF format on your Clarion 5 CD. I keep it installed on the hard drive of my development machines and I have a shortcut to it on the toolbar.

The Embeditor

The IDE also has the Embeditor. With it, you can see your embed code in the context of Topspeed's code. Now you can really see exactly where your code is going to be and how it interacts with Topspeed's.

Where Are Your Priorities?

The IDE has priority settings for embeds also. You can think of these as a mechanism to have "sub embeds" to the regular embeds. And this

gives you the power to insert embedded code between the template generated bits that TopSpeed is using to override the base classes. The call to the parent method is always at priority 5000 so you know your code is before it if your priority is less than 5000 or after it if your priority is greater than 5000. But other methods have even more code and more priority embed points. The `Init` method for a window has priority settings at the Open Files point, Open the window point, and a host of other places.

But I Like Legacy

I know, you like those old familiar embed names. Well, those are still there for you too. While in the embed tree, just press the button that looks like the Clarion for Windows 2.0 icon. Instant legacy embed points! And they are scattered throughout the ABC classes and embed points. This means that now you have an interactive tool that can teach you how your old embed points have changed and what their ABC equivalents are.

What Clarion Isn't

And this brings me to the two quotes at the top of the article. The dream of any Clarion programmer is to be able to embed code anytime, anywhere. This implementation of the IDE, classes, and templates does not do that. The developers at TopSpeed's Development Centre will tell you that they don't want you to get between every little bit of code because you could break something. I feel that with powerful tools, there are always going to be dangerous ways to use them. We need to be given the choice and the power to touch every piece of code.

But there is a conflicting issue. The classes themselves may not be the final version we will see. Already, core portions of ABC have been totally rewritten between C4 and C5. The reason TopSpeed could do this and not break all of our existing code is that the ABC methods have a defined and documented interface. As long as they give you the same interface and embed points, it is very easy for them to change the world around you and never have you miss a beat. But this implies that we don't step in those areas that they want to change.

Is It A TV Dinner?

Clarion isn't a home cooked Cajun meal if you're using ABC and the templates. But, in its current form, it is a far cry from a TV dinner and many miles ahead of Clarion for Windows 2003. Think of it as a care package from your mother. She cooks up some gravy, some soup, a little crawfish etouffee, some gumbo, and she freezes each in a separate container. You can just heat those up when you're hungry and you know it is going to taste better than you could ever make it. And she includes some Cajun sausage and some crawfish tails so you can cook some things for yourself. You can choose to mix and match what you are going to heat up and even cook something fresh yourself to go with it. And it tastes so good.

Lagniappe

I always include a little lagniappe in every article. Lagniappe is that something extra someone gives you that you weren't expecting but that you can enjoy.

This time, it is a jambalaya recipe (pronounced jum buh LI uh) with plenty of embed points. You can choose exactly what you want to change and what you want to go in it. There are suggestions of various things you could use. And there is room for you to add your own ideas.

In This Issue

[Editorial: DevCon Wrapup](#)

Posted on October 20, 1999

[Andrew's Kitchen](#)

Posted on October 20, 1999

[Open Source Update](#)

Posted on October 20, 1999

[ConVic '99](#)

[Clarion Conference](#)

Posted on October 20, 1999

The Anti-TV Dinner Recipe: Chicken and Sausage Jambalaya

Okay. First, here's the recipe in standard form. Let's call this the Clarion ABC Wizard version.

3 cups chicken breasts	1 pound smoked pure pork sausage
2 large onions	1 large bell pepper
2 cloves of garlic	1/2 bunch green onions
2 8 ounce cans tomato sauce	1 can stewed whole tomatoes (Rotel)
2 cups raw rice	salt, cayenne pepper, black pepper
2 tablespoons butter	water

Cut chicken up into bite size pieces. Season with salt, cayenne pepper, and black pepper. Brown chicken in pot with melted butter. Remove chicken. Drain any oil or drippings. Chop onions and bell pepper. Cut garlic finely. Brown onions, bell pepper, and garlic. Add tomato sauce and stewed tomatoes. Simmer 15 minutes. Add chicken and sausage cut into bite size pieces. Cook until chicken is tender. Add raw rice and chopped green onions. Stir ingredients thoroughly. Add enough water to cover ingredients by about 1 inch. Bring to a boil. Cover and cook on low heat for 30 minutes or until rice is tender.

Now for the embed points and overrides...

The chicken breasts are great if you are a non Cajun worrying about your diet. I would substitute 3 cups of cubed pork roast.

Or you could use duck instead of chicken for a richer flavor. You could even use seafood – shrimp or crawfish. Crawfish jambalaya is very good.

I like pure pork sausage because of its rich flavor. There may be reasons that you cannot eat pork (like religious ones) so you could substitute with turkey sausage.

If butter isn't at hand, margarine can be substituted.

Some folks don't like to use tomato sauce or stewed tomatoes. Try 24 ounces of chicken broth along with a little thickening agent like flour or arrowroot.

Maybe you don't have any chicken broth and arrowroot lying around. 3 cans of mushroom soup can suffice just fine.

Now, you've gone from a tomato chicken jambalaya to a crawfish and mushroom sauce jambalaya or a pure pork jambalaya, hold the garlic please!

Add what you like, leave out what you don't. That's the beauty of cooking on your own. You can adjust your recipe at any time. Try that with a TV dinner! And that's exactly what Clarion and ABC gives you: the ability to override or embed your own desired flavor at any point.

Andrew Guidroz II is an active member of the Topspeed community on the Internet and writes software for the insurance/finance industry as well as a host of other industries. His famous Cajun cookouts have become a central feature of Clarion conferences throughout the US. Andrew's Cajun website is <http://www.coonass.com>.

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Feature Article

ABC Design Series The RelationManager Class - Part 2

by David Bayliss

This is a continuation of my [earlier article](#) describing the RelationManager methods. This second part covers aspects of the RelationManager which are not logically related to each other, but neither are they to anything else.

Services

```
ListLinkingFields PROCEDURE(RelationManager Him,FieldPairsClass Trgt, ←
  BYTE RightFirst = 0)
```

This service routine provides the caller with a FieldPairsClass that has been filled with the linking fields of the two RelationManagers (SELF and Him). If RightFirst is zero then the Left of the FieldPairsClass will be filled with fields from the RelationManager denoted by SELF.

In a nice world this code would simply step through the relations, find the relation to Him and copy the Fields.List property into the target element by element (the inner loop). That is almost what happens. The complexity is that only one side of the relation actually stores the field list. So if the code finds that it has a suitable relation but doesn't have the field list it asks the related RelationManager to provide the list, but it has to switch the RightFirst parameter so that the Left/Right fields are correctly oriented in the result.

```
LogoutPrime PROCEDURE,BYTE,PRIVATE
```

Your source for
development tools
and add-ons.

Your outlet for
application sales.

This method is really just an error wrapper around a `PROP:Logout` assignment. This property is used as an alternative to the older `LOGOUT(n,File1,File2,File3,File4...)` procedure call. Using `PROP:Logout` you simply set the property true on all the files you wish to logout before issuing the `LOGOUT` statement. The advantage of this mechanism is that it removes the 52 file limit on the logout and also it means that the files to logout can be selected one by one rather than all needing to appear in one place. This is vital if files are to be switched in and out of the logout to support flexibility of referential integrity.

The code just checks that the file is open, then if logout is required the property assignment is done. An error of 0 means all went OK. An error of 80 means logout is not supported, in which case `UseLogout` is set to zero to prevent the error happening multiple times (this is a programmer error; there is no advantage to informing the user). Any other error is treated as ugly and the user is informed.

Save PROCEDURE,VIRTUAL

This method steps through all the relations and snap-shots the linking fields in the primary into the buffer component of the linking fields buffered pairs class. This is latter used as a sophisticated "record changed" tester for the Update Cascade code.

SetQuickScan PROCEDURE(BYTE On,BYTE Propagate= Propagate:None) ,VIRTUAL

This interesting little method is a variation upon the open/close theme. Essentially it just walks the relation chains using the `Eproc` to ensure it doesn't hit a cycle. Bit 080H of the propagate flag acts as the `Cascading` flag of `Open/Close` fame. The tweak is that this time the caller can specify which type of relations are walked down: one-to-many, many-to-one or all. The work is done inside the loop, and the code uses the fact that a relation is considered many-to-one if this relation has a key to use for a lookup into the other `RelationManager`.

The Plug-Uglies

These routines are all extremely similar. Essentially they all do the same thing: they walk over the relation tree performing some action upon each `RelationManager` they encounter. Before attempting these make sure you are happy with the `Eproc` idea encountered in the [Close method](#). These routines are also all private. They are just too prone to change and re-adjustment to have people rely upon them.

Note that these routines have been simplified in C5EEA (`LogoutDeleteClear` has been removed).

CascadeUpdates PROCEDURE,BYTE,PRIVATE

This is the simplest of the remaining routines so is probably a suitable juncture to explain what all of these routines are doing. `CascadeUpdate` is called when one or more fields in the record buffer for the current file have been changed. The cascading part of the job is to see if any of those fields that are changed are also linking fields to a child file. If they are then the corresponding fields have to be changed (or some other action) in the child file to keep the database consistent.

The code steps through all of the related files checking if there is an RI Update constraint (using `LocalAction`) and then checking to see if the linking fields of these two files have changed since the last call to save. (See [Update](#)) If they have changed then the secondary is told to update itself. If the secondary is unable to do so (that is, an errorcode was returned) then the primary is modified to unchange the fields that were causing the problem. Assuming all the children were changed (if required) then the primary itself is modified to disk.


```
DeleteSecondary PROCEDURE(KEY MyKey,BufferedPairsClass Links,
BYTE Mode),BYTE,PRIVATE,VIRTUAL
```

DeleteSecondary starts by checking that the file is open, and then preserves the contents using SaveFile. This is important, because the person doing the delete expects the primary file to be touched, but may not even know about the secondary file(s) Those files must have their current contents preserved as well (the legacy templates don't bother so be very careful doing RI with legacy templates).

After the file is cleared, the LeftToRight assignment then fills in the linking fields in the child (i.e. this) file with the values from the parent file. Now usually the linking fields will be the whole of the key but it needn't be the case (you can have unassigned key components), so this method uses ClearKey to clear down the remaining key components so that the following SET(MyKey,MyKey) picks up all matching records. (The file clear is not enough, for a descending key clearing down means clearing high!)

Next is a standard ABC sequential processing loop. If the NEXT throws a fatal error (end-of-file is only a notify) then it's necessary to abort the processing, rollback the transaction and get out fast. Well actually not that fast. The exit is always done via a standardised routine that restores the child file to its original state. In less dire circumstances the code checks for two things: hitting end-of-file; or reading a record that no longer matches the parent. In either case all the children have been dealt with so it's possible to return gracefully.

If there is a child then the action depends upon the RI action that has been specified.

- Restrict: In this case the parent cannot be deleted (because there are children) so the method throws an error, rolls back the transaction, and notifies the parent to abort its processing too.
- Clear: This specifies that in the child the link to the parent is blanked out. A simple ClearRight blanks the linking field. Now for the little twist. If you think about it, the children aren't being deleted; they're being modified. Instead of calling DeleteSecondary, the code calls CascadeUpdates.
- Cascade : This is one for the power hungry. The code is very similar to CascadeUpdates. It steps through the children telling them to apply the DeleteSecondary criteria to themselves. Assuming they all manage then it's okay to can delete the parent record with suitable grizzling if unsuccessful.

```
LogoutDelete PROCEDURE, BYTE, PRIVATE
```

This is a tree-walking algorithm in fairly pure form. Its function is to guess all of the files that are likely to be touched by the delete process. It does this by logging out itself, then stepping through each child with an RI Delete constraint. If the constraint is cascade then LogoutDelete is called recursively on the child file (there's no need to worry about cycles as delete constraints only go from 1->Many). If the constraint is clear then the call could be something else but this time it is the LogoutUpdate procedure. See [DeleteSecondary-Clear](#).

```
LogoutUpdate PROCEDURE, BYTE, PRIVATE
```

This method is very similar to LogoutDelete except that it checks the RI Update fields rather than RI Delete. Both cascading and clearing count as modifications so the job can be done by recursing.

```
UpdateSecondary PROCEDURE(KEY MyKey,BufferedPairsClass Links,
BYTE Mode),BYTE,PRIVATE,VIRTUAL
```

And finally, the real nasty one. That said this code is very similar in principle to `DeleteSecondary` (although I doubt I will ever common them up; the task would be just a little too scary). The main changes are:

- The primary record will already have been modified when this routine is called so it doesn't work to call `AssignLeftToRight` to fill in the linking fields. Instead the code uses `AssignBufferToRight` where the buffer has been set up by the preceding `Save` call of the parent.
- If there is a child record then immediately issue a `Save` call on `SELF` (the child). This is to preserve any linking fields to the children (the grand-children of the original record)
- Restrict: As well as aborting if there is a restriction clause the parent record must be modified so that it still points to the children. (`AssignBufferToLeft`)
- Both `Clear` and `Cascade` cases fall down into `CascadeUpdates` which then propagates the changes to the grandchildren.

Summary

So what have you learned? "Never try to read DAB's code!" Well, possibly, but go and have a look at the thousands and thousands of lines of RI code that a decent dictionary generates in legacy templates. Then remember that ABC RI doesn't corrupt your file buffers, can be extended to many-to-many or one-to-one, can cut files in/out of the chain at run time and provides strong [BILF management](#) for free!

In some ways the `RelationManager` heralds all that is good and bad about ABC. It is highly functional, highly efficient, extremely concise, extremely flexible and entirely impenetrable to the casual observer. This brings me to a phrase I used at the '97 DevCon which Steve Parker likes to dispute at all reasonable opportunities: "Don't Know, Don't Care." One of the features of OOP is encapsulation, which means that all of this stuff is safely under wraps. You don't need to understand any of the above, just call the function and have done with it. (You don't even need to call the functions, the templates do this for you.)

Specifically, if you are working at the app level then this is how you should work. However, as my article "[Clarion For Schizophrenics](#)" suggests, all good OOP programmers have a dual nature: one side doesn't know or care, and the other understands the object and can extend it. These design articles are aimed squarely at this alter-ego. As such I trust I have provided it with some food for thought.

[David Bayliss](#) is a Software Development Manager for Topspeed Corporation. He is also Topspeed's compiler writer and the chief architect of the Application Builder Classes.

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

In This Issue

[David Bayliss On The RelationManager - Part 2](#)

Posted on October 26, 1999

[The Novice's Corner: Understanding Clarion Code](#)

Posted on October 26, 1999

[The Clarion Advisor: Detecting Duplicate Records](#)

Posted on October 26, 1999

[October 1999 News](#)

Posted on October 26, 1999

[Product Review: Clarion Source Search](#)

Posted on October 26, 1999



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Feature Article

Understanding Clarion Code: Part 1

by David Harms

In the Clarion world there are generally two approaches to developing applications. The most common is to use the Application Generator for visual design and code generation. The less-used method is to write applications "by hand," using the Clarion language as a 3GL (third generation language).

These seem like two completely different disciplines, but in fact few AppGen users rely strictly on templates for their development. There is almost always some feature in the specification that TopSpeed and/or third party developers haven't thought about or aren't interested in providing, and the only way to solve the problem is with some Clarion code. Usually this code goes into embed points, although in some situations you may see entire procedures written by hand.

Hand code can also be useful for creating small test programs that isolate the code you're working on. Such programs take almost no compile time and make it relatively easy to test and debug since you don't have to deal with the intricacies of the generated code.

One of the best reasons to hand code, however, is to learn how to read and understand source code, whether it's somebody else's source or that created by the AppGen. As Bernard Groperrin has said, everything falls into place when you understand from the bottom up, but it takes a long time to figure it out if you only generate and look at the code.

Whether you write entire applications in source, or just slip a few lines of code into the occasional embed point, the Clarion language offers tremendous power and opportunity. In this first few articles in the series I'll cover some of the fundamentals of writing Clarion code; later articles will deal with more advanced topics.

The Minimal Clarion Program

Your generated applications may have thousands of lines of code (even millions, in the case of legacy apps), but Clarion programs can also be tiny. Listing 1 shows the absolutely minimal Clarion program possible.

Listing 1. The minimal Clarion program.

```
program
code
```



The program in Listing 1 doesn't do anything, which makes it a particularly useless little program except perhaps as a way to win a bet.

A More Realistic Minimal Program

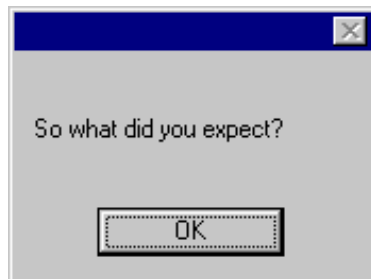
A Clarion program that displays a message and exits is shown in Listing 2.

Listing 2. A slightly less minimal Clarion program.

```
program  
  
map  
end  
  
code  
message('So what did you expect?')
```

This program will display the message shown in Figure 1.

Figure 1. The slightly less minimal Clarion program in action.



The elements of this program are as follows:

1. The program statement indicates the start of a Clarion program
2. A `map/end` statement. This program references a definition of the Clarion `message` procedure, which is defined in `BUILTINS.CLW`. Maps tell the compiler which other modules (source or compiled) are available to this program, and since `BUILTINS.CLW` is almost always needed, the compiler automatically includes it when it sees a `map`.
3. The code statement indicates the beginning of the program's instructions. You need a code statement because almost all programs have data as well, and any data must be declared before the code statement (more on that later).
4. The program instructions, in this case a call to the `message` procedure which displays the specified message (and, by default, an OK button).

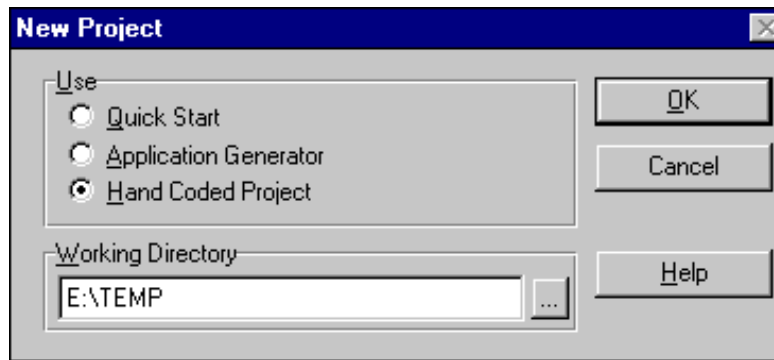
The code in listing 2 is stored in a source file with the extension `.CLW`. In this case the file is called `NOV-B.CLW`. But a source file alone doesn't make a program. The source has to be compiled into object code (not to be confused with object-oriented anything, object code is a binary format), and the object code has to be linked into an EXE or perhaps a DLL or LIB. That's all managed by the project system.

The Project System

The project system supervises the creation of programs (EXEs) or libraries (DLLs and LIBs). The necessary information is stored in either an APP if you're using the AppGen, or a project file if you're hand coding. Project files have the extension `.PR` or `.PRJ`.

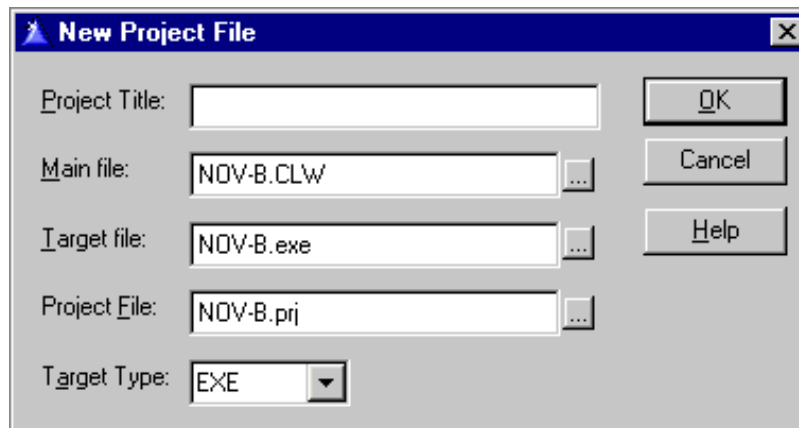
To create a project that will let you compile `NOV-B.CLW` into an EXE choose Project|New from the Clarion main menu. You'll see the dialog in Figure 2.

Figure 2. The New Project dialog.



Choose Hand Coded Project. In the New Project File dialog, fill in the Main File field with the source file name, as shown in Figure 3.

Figure 3. The New Project File dialog.



When you tab off the Main File field the Target file will automatically be set to NOV-B.EXE and the Profile File will be set to NOV-B.PRJ. Click OK. The Project Editor window appears. The Project Editor makes it easy to add new source files, file drivers, or any other necessary components to the application. Click OK to close the project. You can always bring up this window again by choosing Project|Edit.

Despite the fancy project editor interface, project files are just text files. Listing 3 shows NOV-B.PRJ in text form.

Listing 3. The text contained in NOV-B.PRJ.

```
--
#noedit
#system win
#model clarion dll
#pragma debug(vid=>full)
#compile "NOV-A.clw"
#link "NOV-A.exe"
```

This project file specifies that this is a Windows target (there are no other options for Clarion programs), the memory model, the debug setting (in this case "full"), the file to compile, and the application the linker is to create. A more complex project could specify multiple source files, additional compiler pragmas, database drivers, and so forth.

TIP: If you prefer using a text editor to maintain your project files, just give them the extension .PR instead of .PRJ. The Clarion IDE will then open them with the source editor instead of the project editor.

That explains hand-coded projects, but what about applications (APP files)? When you choose Project|Edit when you have an application open, the project editor reads information from inside the APP file itself. Also you don't normally add or remove files from an APP's project – the AppGen handles that for you.

TIP: [Click here](#) for a tip on using PRJ files with APPs.

To compile and run NOV-B, choose Project|Run from the menu or click on the "blue cloud" icon. You can always tell which application will be run by looking at the IDE caption; the current project is listed in parentheses.

Assuming you typed the program correctly, compiling and running the program should bring up the window shown in Figure 1.

Understanding Source Modules

A small program like NOV-B can be more useful than you might think. I frequently create small test programs to isolate bugs or do a proof of concept. They compile quickly and are much easier to edit than an application's embed points.

Real-world Clarion programs are of course much more complex. And you wouldn't want a large program all in a single source file, partly because it could become too large for the editor to handle easily, but more importantly because every change to the source would necessitate compiling the entire program.

A Clarion program can be broken up into procedures, which are logically isolated blocks of code. The code (and data) inside one procedure is not visible to other procedures. You can also place one or more procedures into a separate source module. At one extreme you would have one procedure per source module, and at the other you might have 20 or so procedures per source module (this is very easy to do in the AppGen, and having a number of procedures per module will improve compile times significantly in 32 bit applications).

If you look at NOV-B.CLW (Listing 2) you won't see any procedures just yet. There's just the main entry point defined by the `code` statement. All programs must start somewhere, and in Clarion the source module which has the keyword `program` at its top is the main source module, and execution starts at the `code` statement. Except in small test programs you will very seldom see many statements after this `code` statement. Usually there will be a procedure call, and that procedure, or procedures it calls, will do most of the work.

Listing 4 shows a small program with one procedure which is immediately called when the program starts.

Listing 4. Adding a procedure (NOVC.CLW).

```

program

    map
        MainProc
    end

    code
    MainProc()

MainProc      procedure
    code
    message('So what did you expect?')

```

Now things are getting interesting. The map statement does something besides including BUILTINS.CLW: it declares a procedure prototype which, because it's in a global map is visible to the entire application (you can also have local maps in other source modules, more on that in a later article). The code statement is followed by a call to the MainProc procedure, which is defined at the end of the main source module.

This is almost a model of how many Clarion applications are structured, but not quite. MainProc should really be moved into its own source module, so that only that module needs to be recompiled if the procedure is changed. Listings 5 and 6 show the program split into two source modules.

Listing 5. NOV-D.CLW (the main source module).

```

program

    map
        module('nov-d1.clw')
            MainProc
        end
    end

    code
    MainProc()

```

Listing 6. NOV-D1.CLW (the MainProc source module).

```

member('nov-d.clw')

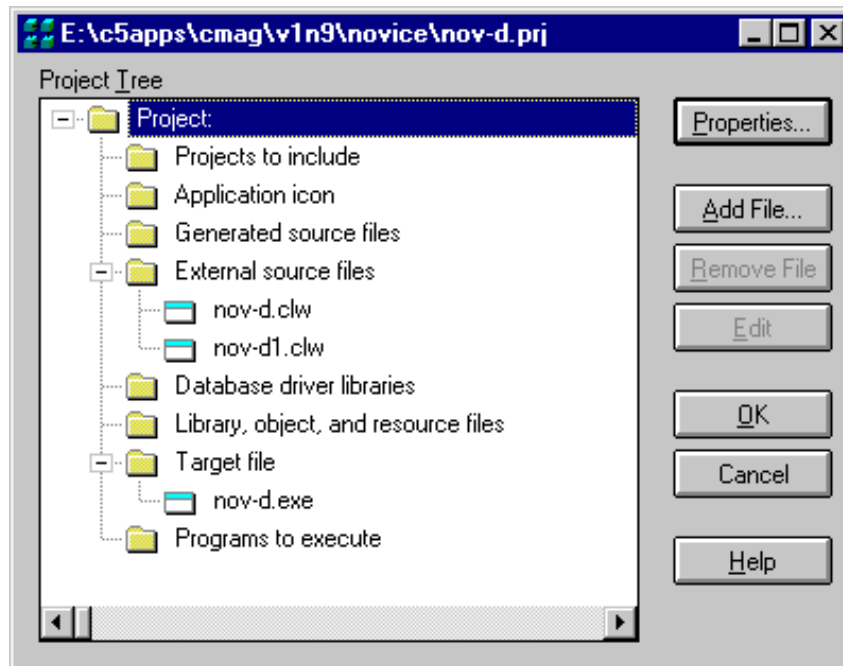
MainProc      procedure
    code
    message('So what did you expect?')

```

The map statement in NOV-D.CLW has now changed, and instead of simply declaring a procedure prototype, it now also declares the module which contains the procedure. That module also introduces something new. At the top of listing 5 you see a member statement. In a multi-module program, all modules other than the one with the program statement must have a member statement. The parameter to the member statement is optional. In generated source the member statement always points to the program source module, which has the same name as the APP file but with the CLW extension.

Because this test program now has two source modules, the project system also has to know about both modules. Simply having a module statement in NOV-D.CLW isn't enough to tell the compiler that you want to compile NOV-D1.CLW. Your project settings should look like those shown in Figure 4.

Figure 4. Project settings for NOV-D.CLW.



Again, if you're working in the AppGen instead of in a hand coded program, the addition of generated modules to the project data is handled automatically.

This small program is starting to mimic how a real application might be set out. The startup code (in the program module) typically calls a main menu procedure, and then depending on what menu options the user chooses other procedures, often in other modules, are called. And those procedures can call other procedures as they choose.

Particularly in legacy applications you will see program execution broken down one level deeper. Inside a procedure you can have a routine, as shown in Listing 7.

Listing 7. NOV-E1.CLW with the addition of an annoying little routine.

```
member('nov-e.clw')

MainProc      procedure
  code
  do BeforeMessage
  message('So what did you expect?')
  do BeforeMessage
  message('Not much of an improvement this time, I'm afraid')

BeforeMessage routine
  message('Click OK to read a message')
```

Routines are much like procedures except they are only visible to the procedure in which they are declared. If you have a block of code you use more than once in a procedure, you're probably better off putting it in a routine and calling it as often as needed. This leaves you with just one place to maintain the code. The same principle, of course, holds true on the larger scale with procedures.

Although this example application is growing procedures and routines, there's still one vital difference between it and any "real" application. A real application will use files, simple data types, and different complex structures. How and where are these kinds of variables declared?

Declaring Data

Traditionally, most high-level programming languages provided global and private variables, which are just what they sound like. Global variables are visible to all parts of the program, while private data are visible only to one part of the program (say, a procedure). The options in Clarion go well beyond these two alternatives.

For starters, variables can be allocated memory one of two ways. Static variables are allocated on the program's heap memory when the program begins, and that memory is not freed until the program stops. Dynamic variables are defined and allocated on the program's stack, typically when a procedure which contains the variable is, respectively, called or created.

Not only can variables be allocated in different ways, they can have different scope. Scope defines which parts of the program can "see" the variable.

- **Global:** Defined after the `program` statement and before the first `code` section in the program module. Global data is visible to the entire program, and should only be used when there's no practical alternative. File declarations and selected variables are normally declared as global. Global data is defined as `static`.
- **Module:** Defined in a non-program module after the `member()` statement and before any procedures. This data is visible to all procedures in the module. Module data is `static`.
- **Procedure:** Defined after a procedure statement but before the procedure's `code` statement indicating the start of executable code. This is visible only to the procedure. Procedure data is dynamic unless it has the `static` attribute. Use `static` on procedure data when you want to make multiple instances of a single procedure share the same variable.
- **Routine:** Variables can be defined within a routine, visible only to that routine, but you must use the `data` statement after the routine name, then declare the variable(s), then use a `code` statement before the start of the routine code. This capability is a byproduct of the compiler internally treating routines as procedures.
- **Threaded:** This is an attribute of variables which specifies that a new buffer is created for each thread of execution. (I'll deal with

threads in a later article.)

- **Class:** Data scoping in classes is more complex than for procedural code, and beyond the scope of this article. See the [ABCs Of OOP series](#) for more information.

Related to data scoping is the use of the member statement. If a module has a member statement that points back to the program module, then it can see any global data declared in the program module. If the member statement has no parameter then that module can only see data declared inside that module (or data referenced through an include statement, which will be covered in a later article).

Listings 8 and 9 show the program and member module for a small program that demonstrates some of the scoping rules.

Listing 8. NOV-F.CLW with global data and two procedures.

```
program
  map
    module('nov-f1.clw')
      MainProc
      TestProc
    end
  end
  GloVar      short

  code
  MainProc()
```

Listing 9. NOV-F1.CLW with module, procedure, and routine data.

```
member('nov-f.clw')
!-----
! Module data is visible to all the procedures and
! routines in this source module.
!-----
ModVar      long

!-----
! MainProc procedure
!-----
MainProc    procedure
ProcVar     byte  ! Visible only to the procedure
  code
  TestProc()
  do SetValue
  message('MainProc: ProcVar = ' & ProcVar & ', |
    ModVar = ' & ModVar & ', GloVar = ' & GloVar)

SetValue    routine
  ProcVar = 7

!-----
! TestProc procedure
!-----
TestProc    procedure
```

```

code
do SetValue

SetValue      routine
data
RtnVar        byte  ! Visible only to this routine
code
RtnVar = 234
ModVar = 123322
GloVar = 8998
message('TestProc: Routine variable = ' & RtnVar |
& ', ModVar = ' & ModVar)

```

I'll have more detail on declaring variables in the next article in this series. For now, you can see that I've used the integer types of byte (one byte unsigned), short (two bytes signed), and long (four bytes signed). Also note that I've included some comments in the text; comments are always preceded by an exclamation point.

There are several other features of interest. Each procedure has a routine called SetValue, but because the routines are in different procedures and not visible to each other there is no name clash. Also one of the routines uses the traditional format, whereas the other declares some routine data.

In several of the message statements I've used the vertical bar (|) continuation character to split the line, and the concatenation operator (&) to create a message out of string literals and variables.

Good Programming Practice

As a rule of thumb, the more global data your application has, the more prone it will be to bugs. Certain kinds of data, at least at present, pretty much have to be global. This includes file declarations and a small number of variables.

The problem with global data is the same as the benefit of global data: any code, anywhere in your application, can not only read the value of the data, but set it. If you lose track of which procedure is calling which other procedure, you may discover that something you thought had one value now has another value. For a detailed discussion of the kinds of problems this can create, read David Bayliss's article entitled "[Propitious Memory Corruption](#)."

In general, you want to restrict the scope of your variables as much as possible. If you have a variable that's only needed in a routine, declare it there. The same goes for procedures and modules. If you really do need to share data between procedures you should also consider passing the data as a parameter, which is one of the subjects I'll deal with next month.

Summary

Even if you're a hard-core AppGen user, the Clarion language has much to offer. Whether you want to sort out a bug, prototype a new feature, or just have some fun learning, it's easy to get started.

If you have any questions about the terms and/or concepts discussed in this article, please notify me by email or, preferably, by posting a message in the [clarionmag.features newsgroup](#).

[Click here to download the source](#)

[David Harms](#) is an independent software developer and the co-author with Ross Santos of Developing Clarion for Windows Applications,

In This Issue

[David Bayliss On The RelationManager - Part 2](#)

Posted on October 26, 1999

[The Novice's Corner: Understanding Clarion Code](#)

Posted on October 26, 1999

[The Clarion Advisor: Detecting Duplicate Records](#)

Posted on October 26, 1999

[October 1999 News](#)

Posted on October 26, 1999

[Product Review:](#)

published by SAMS (1995). He is also the editor and publisher of Clarion Magazine.

[Clarion Source Search](#)

Posted on October 26, 1999

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



Clarion magazine

[Main Page](#)
[Log In](#)
[Subscribe](#)
[Open Source](#)
[Links](#)
[Mailing Lists](#)
[Advertising](#)
[Submissions](#)
[Contact Us](#)
[Site Index](#)
[ClarionMag FAQ](#)
[Download PDFs](#)
[Search ClarionMag](#)

Feature Article

Detecting Duplicate Records

by Gordon Smith

Recently on the newsgroups someone mentioned that TopScan didn't identify duplicate record errors while building the keys. After digging around in the source code I confirmed this to be indeed true and set about writing a "duplicate record" detector (DRD) which was file neutral. This article will take a look at the resulting class.

The Requirement

The DRD would have to perform the following functions:

- Locate all duplicate errors, for each key.
- For each duplicate error, locate all clashing records (remember each duplicate error can have more than two associated records)

The Solution Overview

The solution ended up being quite a bit simpler than I had originally envisaged, although on the down side it involves a brute force approach where every record in the file will need to be tested. The logical flow goes something like this:

1. Rebuild all keys (this is essential, as keys must be up to date to enable location of clashing records).
2. Loop through each record in file sequence.
3. For each record use the `DUPLICATE(FILE)` function to see if it is a duplicate.
4. For each duplicate record, loop through all the keys to find which ones are reporting clashes (remembering that it can be more than one key causing the error).
5. For each duplicate key, find the one valid record and associate the current (error) record with it.

It is important to note that when duplicate records exist in a given key there will always be one valid record, while the rest of the duplicates will simply not exist for that key.

There are several ways a duplicate record can be created, the following are probably the most popular:

1. APPEND: Since append doesn't update any keys it is probably the easiest way to create duplicate records.
2. File conversion (common type A): If the conversion utility simply creates a new empty file and appends all the records from the "before" file, then this will be the same as 1.
3. File conversion (Common type B): It is possible in some file drivers to modify a key structure (remove a DUP attribute for example) and then simply delete the key (somefile.k01 for example) from the hard drive, allowing Clarion to rebuild it afterwards.

The Solution



The solution has been presented in the form of a class. (The attached source code also has a small example PRJ and CLW file). The definition looks something like this:

```
FindDupClass      class, type
init              procedure(file f)
kill              procedure
buildKeys         procedure
findDup           procedure
display           procedure
end
```

The class would typically be used as follows (and since it is a class, the actual code is appropriately simple):

```
TestFileForDup procedure(FILE TestFile)

cFind FindDupClass

code
cFind.Init(TestFile)
cFind.BuildKeys()
cFind.FindDup()
cFind.Display()
cFind.Kill()
```

The code "notables" now follow:

BuildKeys Method

Rather than calling `BUILD(FILE)`, this method builds each key individually by using the `PROP:Keys` and `PROP:Key` file properties. The main reason for this is to avoid a nasty side effect of `BUILD(FILE)` where it aborts BUILDING all keys when an error is encountered (in this case a duplicate record error!). A nice addition to this method would be to actually check if any errors occurred during this operation, if not then there will be no need to check for duplicates. Another useful addition would be to use the `PROP:Completed` and `PROP:ProgressEvents` properties to display a nice progress window, with the option to gracefully cancel.

```
FindDupClass.BuildKeys procedure

i unsigned, auto
k &key

code
loop i = 1 to self.f{PROP:Keys}
  k &= self.f{PROP:Key, i}
  build(k)
end
```

FindDup Method

The `FindDup` method is the brute force part of the solution (it loops through the entire file, record by record). Since files can be large it uses the `EVENT:Timer` event on a simple status window. When a duplicate is found, a private method is called (`CalcDupInfo`) to analyse the duplicate.

```
FindDupClass.findDup procedure
```

```
Prog long(0)
```

```
Found long(0)
```

```
Window WINDOW('Caption'),AT(,,200,44) ←
  ,FONT('MS Sans Serif',8,,FONT:regular) ←
  ,TIMER(1),SYSTEM,GRAY,DOUBLE
  PROGRESS,USE(Prog),AT(5,10,190,10),RANGE(0,100)
  PROMPT('Duplicates:'),AT(5,27)
  BUTTON('&Cancel'),AT(150,25,45,14),USE(?ButtCancel)
  STRING(@s64),AT(50,27),USE(Found)
END
```

```
i unsigned, auto
```

```
FinFlag byte(FALSE)
```

```
code
```

```
open(self.f)
```

```
assert(~errorcode())
```

```
stream(self.f)
```

```
assert(~errorcode())
```

```
set(self.f)
```

```
open(window)
```

```
window{PROP:Text} = self.f{PROP:Name}
```

```
?Prog{PROP:RangeHigh} = records(self.f)
```

```
accept
```

```
case event()
```

```
of EVENT:Timer
```

```
if ~FinFlag
```

```
loop 100 times
```

```
next(self.f)
```

```
if errorcode()
```

```
FinFlag = TRUE
```

```
post(EVENT:CloseWindow)
```

```
break
```

```
end
```

```
Prog += 1; display(?Prog)
```

```
if duplicate(self.f)
```

```
Found += 1; display(?Found)
```

```
self.CalcDupInfo()
```

```
end
```

```
end
```

```
end
```

```
end
```

```
end
```

```
close(window)
```

```
close(self.f)
```

```
return FinFlag
```

calcDupInfo Method

This method goes through the following steps to find duplicates.

1. Save the current pointer and position for the current record.
2. Check each key to find the ones reporting the error.
3. For each duplicate record there will be one valid entry. This is located using `GET(SEL.F,K)` procedure. Since the current record buffer for the "error" record "matches" the one valid record, when the `GET` is called it will return the one valid record!
4. For each duplicate `AddResult` is called twice (NOTE: `AddResult` will only add records to the result queue if they haven't already been added):
 - a. To add the valid record to the result.
 - b. To append the duplicate record to its associated valid record.

The original record is then restored with the `RESET, NEXT` combination (you must use the `RESET(FILE)` form of `RESET`, for obvious reasons).

```
FindDupClass.calcDupInfo procedure
```

```

j unsigned, auto
k &key
tmpPos like(ResultQueue.Pos)
tmpPoint long, auto

code
tmpPos = position(self.f)
tmpPoint = pointer(self.f)
loop j = 1 to self.f{PROP:Keys}
  k &= self.f{PROP:Key, j}
  if duplicate(k)
    get(self.f, k)
    assert(~errorcode())
    self.AddResult(pointer(self.f), k)
    self.AddResult(tmpPoint, k, TRUE)
    reset(self.f, tmpPos)
    next(self.f)
  assert(~errorcode())
end
end

```

AddResult Method

This method adds the duplicate record information to the result queue for displaying later (in the display method). It checks that this particular duplicate record hasn't been added already (based on the key name and its `POSITION` information), and if it has it will append the current record number to it.

`FindDupClass.addResult procedure(string pnt, key k, byte AppendPos = FALSE)`


```

code
clear(self.qResult)
self.qResult.Rec = pnt
self.qResult.Key = k{PROP:Label}
self.qResult.Pos = position(k)
get(self.qResult, +self.qResult.Key, +self.qResult.Pos)
if errorcode()
  add(self.qResult, self.qResult.Rec)
elseif AppendPos
  self.qResult.Rec = clip(self.qResult.Rec) & ', ' & pnt
  put(self.qResult)
end

```

Summary

All in all it was quite pleasing that the solution ended up being so straightforward. Something similar will undoubtedly find its way into TopScan. One more point: to verify the sample program, try using TopScan to view the created file.

[Download the source](#)

Prior to joining TopSpeed Development Centre, Gordon Smith worked for an Irish company developing software for multi-national pharmaceutical companies. He was also a member of the 3rd party accessories program (Compile Manager 2) and developed the Clarion Class Browser.

In This Issue

[David Bayliss On The RelationManager - Part 2](#)

Posted on October 26, 1999

[The Novice's Corner: Understanding Clarion Code](#)

Posted on October 26, 1999

[The Clarion Advisor: Detecting Duplicate Records](#)

Posted on October 26, 1999

[October 1999 News](#)

Posted on October 26, 1999

[Product Review: Clarion Source Search](#)

Posted on October 26, 1999

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Clarion News

October 26, 1999

[Linder SetupBuilder 3.0 Beta 6 Available](#)

Linder SetupBuilder 3.0 Beta 6 is available now. The installation is password protected. If you are already a SetupBuilder beta tester but you did not receive a Beta 6 notification or wish to beta test email ceo@lindersoftware.com. Special beta pricing of \$71.40 USD ends Oct 31, after which the price goes up to US\$119.00.

[New Clarion Addons](#)

Solid Software is releasing some new third party products including MessageEx, a MESSAGE() function enhancement, as well as selected freeware. More to come soon.

[App-Ref Version 2.00 Released](#)

Sable Software has announced the release of App-Ref 2.0, a Clarion cross-reference tool. New features include: improved detection of variables and files using the ABC dot syntax; zoom button to view where variables are used in source; and global template search. A 30 day trial is available. If you have looked at App-Ref before, Version 2.0 will give you a new 30 day test period. \$145.00 Australian direct from Sable Software or purchase from <http://www.clarionshop.com> for US\$95.

October 20, 1999

[Y2K Compliant DATE Function](#)

Philip Will of ProDomus has provided a free replacement Clarion DATE function which correctly handles certain leap year dates (i.e. DATE(14,29,1998) will return March 1, 2000 rather than February 28). This bug should be fixed in the next Clarion release. In the meantime this function can be implemented by renaming the BUILDINS.CLW DATE function and recompiling. Use at your own risk.

[Cowboy's DevCon Presentation](#)

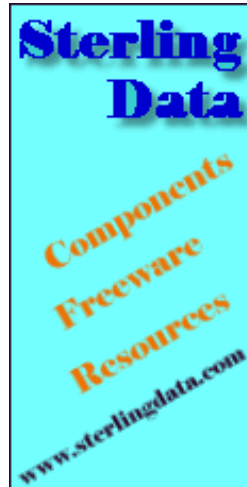
Andy Stapleton's DevCon '99 presentation (DEVCON99.ZIP) is now available for download. This will show you step by step how to convert to MS SQL 7.

[RAD Accounting Source Code Price Cut](#)

Mitten Software is now selling RAD accounting source code for \$99. This includes support for both Legacy and ABC templates. RAD Messenger is now \$49, for legacy templates only. RAD Accounting is a comprehensive modular Accounting System designed in Clarion. It consists of General Ledger, Accounts Receivable, Accounts Payable, Job Costing and Bank Reconciliation.

[Clarion Source Search Beta 2](#)

Beta 2 of Clarion Source Search now makes context sensitive calls to Clarion Help files the same way the Clarion environment does. There are new label and line search features, and the file viewer colors have been enhanced for templates, STY, and TXA files. Editor calls now pass line numbers. The search engine now also has some support for C and CPP files.



In This Issue

[David Bayliss On The RelationManager - Part 2](#)

Posted on October 26, 1999

[The Novice's Corner: Understanding Clarion Code](#)

Posted on October 26, 1999

[The Clarion Advisor: Detecting Duplicate Records](#)

Posted on October 26, 1999

[IFT Server Autumn Sale](#)

The IFT Server templates are on sale for \$129. IFT Server makes it easy for you to make your own royalty-free internet application servers. Includes the PowerMerge templates for building server-side includes. The \$20 discount still applies if you also buy the IFT Client templates at \$89 for a total of \$198. [Read a review](#) of IFT Server templates.

[IFT ShapeMaker Radically Changes Application Appearance](#)

Coming soon: Jazz up your application's windows with rounded corners, or even create ellipse-shaped windows. System menu functionality is retained. Demo now available.

[Free TrashFlash Template](#)

Sterling Data's free TrashFlash template lets you drag delete browse items by dragging/dropping to a trash can. TrashFlash is also the subject of Part 2 of the tutorial "Template Programming for Beginners", where the template is explained line-by-line.

[Tintools Updated](#)

The final version of Tintools for Clarion 4 and 5 is now in release. Starting with version 5.1.0 of Das tools for Clarion 5, Tintools templates and functions such as the report viewer are being merged into the free portion of Das Tools. Both versions are available for download.

[Tom Hebenstreit's DevCon Presentation Available For Download](#)

Tom Hebenstreit's DevCon presentation ("Building Distributed Systems Using Email") is now available for download. Included are links to useful sites for learning Internet email standards, 3rd party vendors mentioned in the talk, the PPT slides, some sample source code and various other useful bits.

October 14, 1999

[TopSpeed Releases Intellidate Patch For Clarion 2.1](#)

TopSpeed has released a patch which adds Intellidate capabilities to Clarion 2.1, to address Y2K compliance for applications using two digit date pictures.

[New Developers Photo Gallery Feature](#)

Mike McLoughlin has added the option of marking photos in the Clarion Developers Photo Gallery with an asterisk to show who is currently looking for work. Photos of developers not looking for work are of course still welcome!

[ARCO Word Reporter Update](#)

An ARCO Word Reporter update (Version 2.4) is now available. Includes fixes to some problems with Word tables.

[DeveloperPLUS Grand Opening](#)

DeveloperPLUS is marking the grand opening of its Clarion Third-Party Store with a 15% discount on all Lodestar and Cowboy Computing Solutions products through the month of October. Products come with a 30 day money back guarantee.

[ForKeeps v.4.1 Released](#)

Version 4.1 of the ForKeeps message archive has been released. New features include sort orders in search results, changes to tab orders and list views, a section view for redirected newsgroups, option to disable the logo wallpaper, user-defined sender/receiver fields format, and better Eudora export. The upgrade is free to 4.x users and all who purchased in or after October 1988, \$20 for the upgrade from any older version.

[CollegeAid Calculator 2000-2001](#)

The CollegeAid Calculator 2000-2001 (subject of a [Clarion developer profile](#) earlier in Clarion Magazine) is available for download. This is a comprehensive college financial planning tool. \$10 by download, \$25 on disk.

[Sebastian Talamoni's DevCon Examples Available By FTP](#)

Sebastian Talamoni's DevCon examples are now available for download

[October 1999](#)

[News](#)

Posted on October 26, 1999

[Product Review: Clarion Source Search](#)

Posted on October 26, 1999

from UniSoft. The ABC session zip includes the PowerPoint presentation, Libsrc directory, and example app; the wizatron session zip includes the PowerPoint presentation and the example styles.

October 6, 1999

[New Third Party Marketer](#)

DeveloperPLUS is a new service for Clarion third party publishers offering a number of different levels of participation. All accounts include real-time credit card processing and email notification of purchases, and dedicated product order forms. Rates vary at four levels from 4.5% of product price + \$2.00 USD (Basic) to 10.5% of product price + \$4.00 USD (Enterprise), per transaction. Enterprise accounts receive catalog space, real-time product fulfillment, FTP storage, demo and freeware cataloging, custom product unlock code generation, and more.

[ProDomus Translator Plus In Beta Release](#)

ProDomus is pleased to announce the release of Translator Plus, a set of internationalization development tools, made up of ABC compliant classes. Components include picture, string, environment, character, and type classes, and utility components manage and speed the internationalization process by extracting labels and text strings and assisting with translation.

[Dalby Source Printer Now Available](#)

The Dalby Source Printer v5 is released now and available for download. Changes in version 5 include: fixes for Clarion 5; 32 bit program with support for long filenames; option to print only embed code; and improved embed headers. Cost: \$80.

[Linder Software Looking For Wise Scripts](#)

The forthcoming Linder SetupBuilder 3.0 (Beta 6) comes with a new Wise Script Converter. This converter is able to read in your Wise scripts and convert to the SetupBuilder project format. Linder is looking for some Wise scripts for testing. Please do not send the installation executables!

[Mike Pasley's DevCon Presentation Available](#)

Mike Pasley's DevCon PowerPoint presentation on Function Point Analysis is now available for download. On the [web site](#) select the PlanetClarion menu item and then the section on Function Point Analysis. While in this section sign the FPA guest book if you would like updates.

[Read the September 1999 News](#)

[Read the November 1999 News](#)

Do you have a news story or press release we should know about? Send it to editor@clarionmag.com

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Product Review

Clarion Source Search v1.5 from Carl T. Barnes

by [Tom Hebenstreit](#), Reviews Editor

Where is it?

Where did I declare that variable, that file? Where did I use it? Where is this function used? Which DLL? What was the name of the procedure? What embed was that used in?

WHERE THE \$(#! IS IT??

Sound familiar to you? It sure does to me.

While CW is great at generating code and organizing a program at the higher levels, it's also great at letting you hide bits and pieces of code in a million little nooks and crannies such as embeds, data buttons and template prompts. Finding all usages of a particular variable or string, for example, can be a major pain in the, er... mouse, especially when you are dealing with hundreds of procedures scattered across multiple apps.

This is where Clarion Source Search (CSS) enters the picture. What does it do? Just what the name says – it searches source code and displays a list of results. Actually, it will search any ASCII type file, so it will happily inspect CLW, CLA, INC, TPL, TPW, INI, EQU or any other plain text file.

So, you say, what makes it so different from any of a hundred other text search programs? Why should you consider buying a search program when the built-in Windows 'Find' command can search text files for you?

CSS understands Clarion. It knows about labels, modules, procedures, routines, classes, methods and all the rest. When it finds a match, it not only shows you the line where the item occurs, it shows you the label of the procedure, the routine label (if applicable) and more. And that helps you put those bits and pieces back into perspective.

Doesn't sound like enough? I have to confess that I pretty much greeted CSS with shrug when Carl Barnes (the author) first sent a beta to me. Searching text, ho-hum. Once I began really using it though, I started singing an entirely different tune...

Installation

To start at the beginning (always a useful thing to do), I downloaded the latest CSS from Carl's web site as a single file installation program. A few clicks later and I was done; CSS was installed and ready to go. I only have one major suggestion, and that is (as usual) that the installer should display the Help file at the end of the installation process.

By the way, CSS is purely a standalone utility. There are no templates or other files required to use it, and it works happily on source code from any version of Clarion: Windows or DOS, legacy or ABC, whatever.

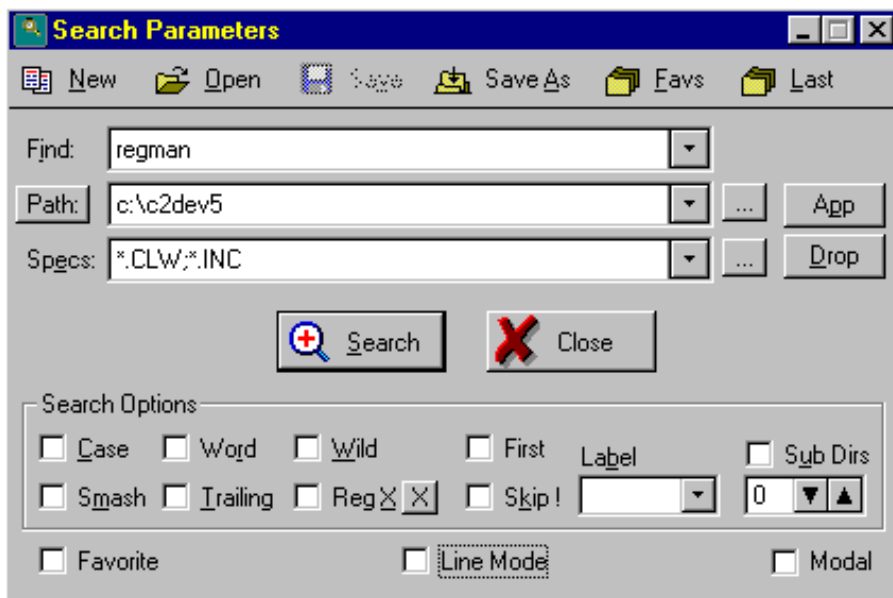


CSS in action

On the theory that a picture is worth 1995 words (inflation, you know), let's take a look at CSS at work.

To begin with, using CSS is dead easy. You tell it what to look for, it finds it for you. Thus, everything logically starts from the Search Parameters window where you tell it what to search, what to search for and how to search.

Figure 1. The CSS Search Parameters window.

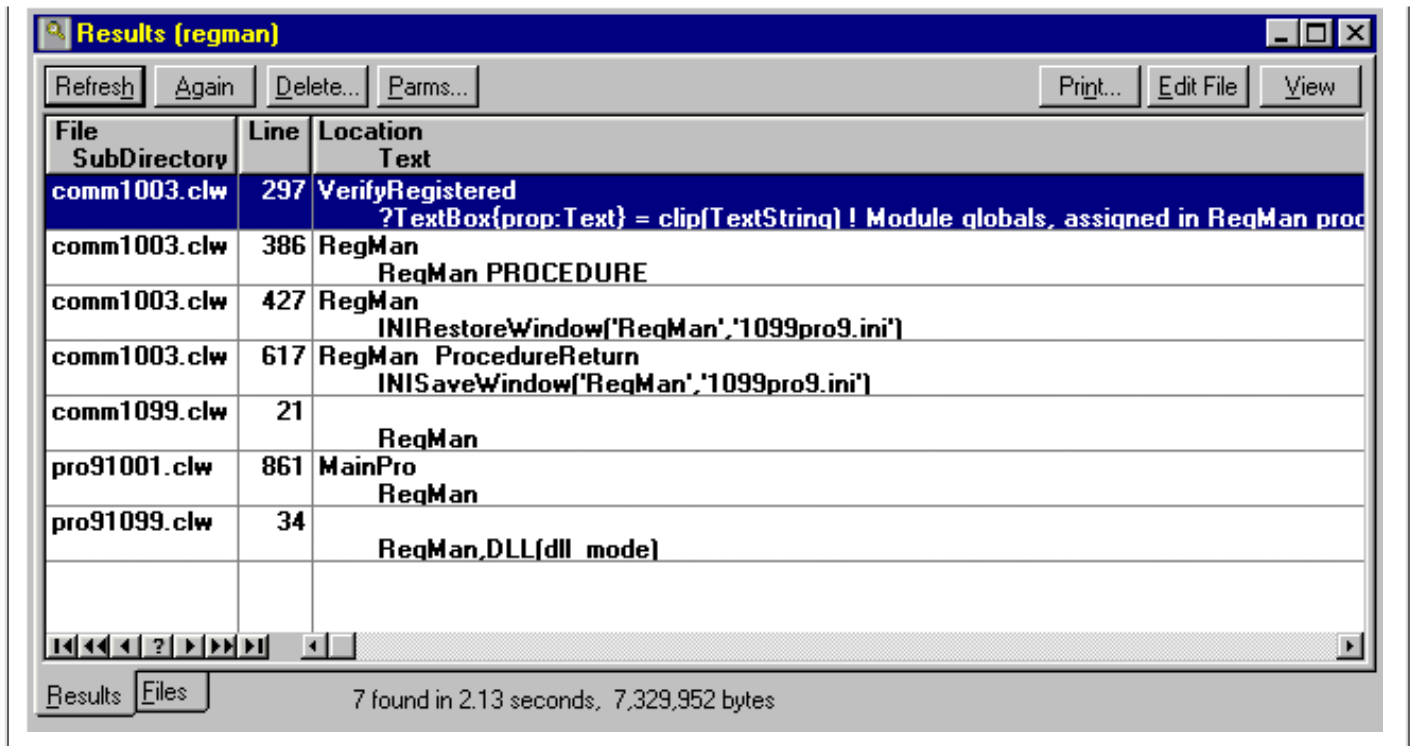


(Lots of options, eh? You'd think that Carl had to pay for every vowel he put on the screen the way some of the choices are abbreviated. I guess he had to fit them in somehow. But I digress...)

Some of the options I could choose to use include Case (case sensitivity), Smash (remove extra spaces from results), Word (must be full word), Trailing (text must end with a space), Wild (use standard * and ? wildcards), RegX (use Regular Expressions – more on this later), Skip ! (don't search comments) and others.

In this case, I have pretty much just accepted all of the default search options after telling it where to look and what to find (the name of a procedure called "RegMan"). Here are the results:

Figure 2. Results of a search for a procedure named RegMan

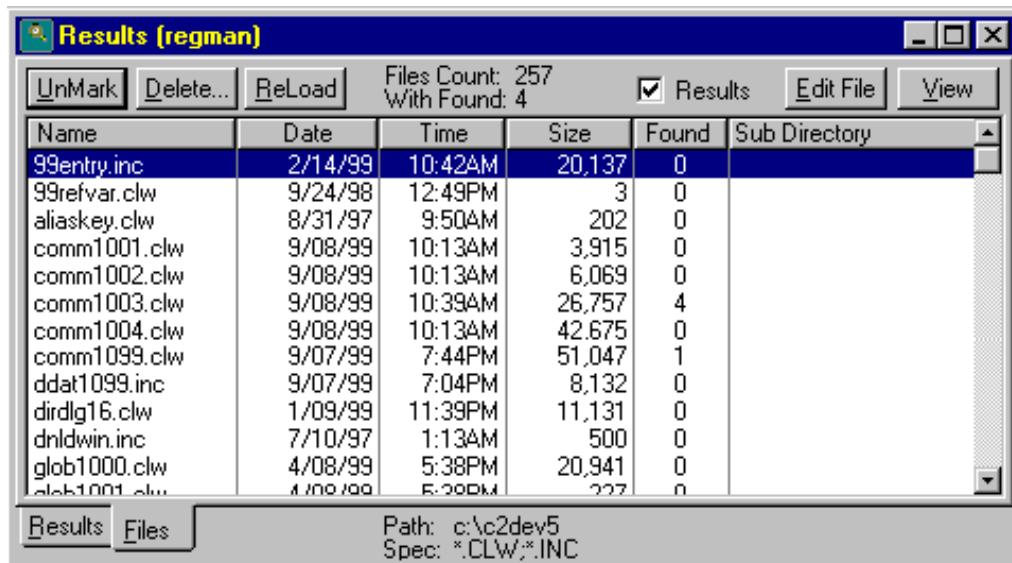


CSS found 7 occurrences in 2.13 seconds after searching 257 files (totaling around 7 megabytes). If you look at the list, you'll see that:

- the first occurrence was in a comment in another proc called VerifyRegistered, in source module comm1003.clw (it would have been eliminated from the list if I had used the Skip ! option)
- the second is the definition of the RegMan procedure label itself
- the third is inside RegMan, restoring window INI values
- the fourth is inside RegMan in the ProcedureReturn routine where it is saving the window INI values
- the fifth (and seventh) are the procedure prototypes in app where it resides and as an external DLL procedure
- the sixth is where the procedure is actually called from within a procedure called MainPro

Next, note the Files tab on the bottom of the list. Clicking on that brings up the list of source files that were searched:

Figure 3. A few of the 257 files that were searched.



Here you can see two of the files that contained the string (the first had four occurrences, the second had one). What if I want to get rid of the other 253 files that didn't contain RegMan? No problem, click on the Delete... button, then select the Found count = 0 option from the menu. Your file list is now just the four files.

Want to go deeper? From both the Figure 2 or the Figure 3 list, you can double-click on an item and you will then see the actual source code with various result lines color-coded to make them stand out.

Figure 4. The CSS source code viewer.

```

View comm1003.clw - line 617 - (4 of 7)
<< Prev Next >> As Text Edit File Res Break Find CW?
RegMan ProcedureReturn
if SaveWindowPositions = 2
  INISaveWindow('RegMan','1099pro9.ini')
end
CLOSE(window)
END
IF LocalResponse
  GlobalResponse = LocalResponse
ELSE
  GlobalResponse = RequestCancelled
END
POPBIND
RETURN
-----
InitializeWindow ROUTINE
?!
?! This routine is used to prepare any control templates for
?! per procedure.
  
```

Check out how the line with the search string is green, while the label for the next routine is highlighted in gray to make it easier to scan the code.

From this screen, you also have options to go to the next or previous file, the next or previous occurrence of the search string, to view the code as text (not highlighted, but you can cut and paste), to edit the source, to search the displayed source file for another string, and more. Using the CW? button, you can invoke Clarion language help for a selected keyword.

Is there more? You bet – there are a ton of other options and switches that you can use to filter, print and otherwise muck about with your searches. Here's just a part of the list from the CSS web site:

- Search specs can be saved and recalled to help start common searches fast
- Favorite searches list allows recalling most common search specs in two clicks
- Last 100 complete search specs saved for review and reuse
- Filter search results (compound search) nine different ways including complex wildcard and regex (Regular Expressions)
- Regular Expression assistant helps you learn and write and use Regex
- Find the declaration of equates, classes, files or procedures fast with a label option
- Write search results to an ASCII file for saving or emailing
- Multi-threaded - Entry, Results and Viewer all run in separate threads for maximum flexibility

- Configuration options allow customizing CSS look, feel and function as you desire
- Drag and drop paths from Windows Explorer

Apart from the straight-ahead uses I've already mentioned, CSS can be a great tool for learning ABC or the template language. For example, let's say you are coding along and find that you need to know exactly what is going on in the `FileManager.Open` method. Search the Clarion 5 LibSrc folder for the string "FileManager.Open". In a couple seconds, you'll see it is declared on line 845 of `AbFile.clw`. Double-click, and you see that all it does is call another method called `FileManager.OpenServer(1)`. Well, that happens to be listed two items above on the search results list, so you can pop right over there with a click or so. Within that method, you can see calls to `SELF.OpenFile`, the third `FileManager` method that appeared on the results list. One click and you are there. Way cool, as they say.

Regular Expressions

Regular Expressions (commonly abbreviated as "regex") are a way of creating what you might think of as super wildcards. They allow you to create expressions (mini-formulas, if you will) that allow for very flexible pattern matching. You can even create expressions that test for not having particular characters or strings (try that with wildcards!). A simple example used in the CSS help file is one which would find all instances of a HLP (help) attribute which don't have a tilde in them:

```
HLP ( '[^~]
```

Do you think it looks way too complicated to figure out how to use regex? Never fear, CSS has a Regular Expression Assistant that lists the various regex operators and allows you to test variations of your expression on the fly against sample text. Using regex just doesn't get any easier than this, and it adds another really powerful dimension to CSS's search capabilities. (NOTE: You can get a bit more information on Regular Expressions in the Clarion 5 help topic for the `MATCH()` statement.)

Performance

CSS is fast. No, make that very fast. On my laptop (a 400MHz AMD K6), CSS routinely scans anywhere from four to ten thousand lines of code per second when simply searching for a string or using standard wildcards. Using a Regular Expression that found several hundred occurrences of the search string slowed it down to a mere three to five thousand lines per second. On a desktop machine with fast SCSI drives, it searched up to thirty thousand lines per second.

Let's just say that you won't be getting any cups of coffee while waiting for CSS to do its thing.

Documentation

Documentation is provided in the form of a standard Windows Help file. It does a good job of explaining both the theory and practice of using CSS, but the best part is that you don't really need it until you decide to start exploring the more advanced features such as Regular Expressions.

As I mentioned above, good tool tips are also provided for all options. With their assistance, the program is simple enough that you can just pretty much jump right in and start using it.

Technical Support

Support is good, with most email questions being answered within a day (or at most, a couple days). In months of use, though, I have yet to have a question that I really needed to ask.

Suggestions? Now that's another matter (I've always got a million of 'em). CSS had, umm, how shall I say it... a rather lengthy beta period (over a year). I'm pleased to report, though, that both during the beta and since the release Carl has been great about incorporating

In This Issue

[David Bayliss On The RelationManager - Part 2](#)

Posted on October 26, 1999

[The Novice's Corner: Understanding Clarion Code](#)

Posted on October 26, 1999

[The Clarion Advisor: Detecting Duplicate Records](#)

Posted on October

suggestions for improvements into the product.

26, 1999

Summary

You know, the most important commodity we have (and the one we always seem to be shortest of) is time. And nothing can waste time like searching through piles of files trying to find every occurrence of something (or dealing with that nagging feeling that you just might have missed one). Or trying to find one ABC method out of hundreds. Or that file, or... I could go on and on, but I won't. It's just hard to exaggerate how really useful this utility is once you start using it.


The bottom line? For myself, I won't start Clarion without CCS.





[October 1999
News](#)

Posted on October
26, 1999

[Product Review:
Clarion Source
Search](#)

Posted on October
26, 1999

PRODUCT RATING	
Overall	
Ability to do the task	Excellent
Ease of use	Very Good
Ease of installation	Good
Documentation	Good
Technical support	Very Good
Black box DLLs/LIBs	N/A

LEGEND	
First class all the way	
More than adequate	
Barely adequate	
Don't even think about it	

Clarion Source Search v1.5 lists for US\$45.99

CSS is available directly from Carl Barnes via BMT Micro using secure online ordering. If you prefer, you can also order by phone.

Instructions for ordering, more information and a downloadable demo can all be found at his website: <http://www.carlbarnes.com>

Check it out!

Copyright © 1999 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. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.