Clarion Magazine

Clarion News

- » ClarionLive Weekly Webinar #24 - Robert Paresi on iQ-Sync and Bruce Johnson on OddJob and WinEvent
- » iQ-Sync
- » LogFlash Half Price
- » Aussie DevCon Training Schedule
- » Handy Zip'N Email and Handy Zip'N FTP
- » Product Scope 7.9
- » CommandBars Wrapper Template 2.00
- » EZChangeLog Reporter 1.7
- » CHT Build 13C1.00

[More news]

- » Clarion.NET FAQ
- » Clarion# Language Comparison
- » Proper Case in Clarion#
- » Event Handling In Clarion#
- » Code Snippets In The New IDE
- » Understanding Clarion# Events
- » Understanding Clarion# Delegates

[More Clarion & .NET]

- » Clarion 101: Creating C7 Dictionaries
- » Clarion 101: Importing Apps Into Clarion 7
- » Clarion 101: Understanding The C7 IDE (Tools | Options)
- » Clarion 101: Understanding The C7 IDE
- » Embed Analysis Part 3

[More Clarion 101]

Latest Free Content

- » The Clarion Magazine Roadmap
- » Source Code Library 2009.06.30 Available
- » Product Review: Report and Presentation Manager (RPM)

Latest Subscriber Content

PDF for July/August 2009

All articles for July/August 2009 in PDF format.

Posted Monday, August 31, 2009

Topspeed Superfiles

TopSpeed superfiles, which allow you to combine multiple TPS "files" into a single physical file, have a curious reputation. Some developers love the, others won't touch them. Steve Parker explains the pros and cons of superfiles.

Posted Friday, August 28, 2009

The C7 Experience

After building his first from-scratch C7 application, Steve Parker takes a step back and looks at the experience of doing new C7 development.

Posted Tuesday, August 11, 2009

Creating An XML RSS Web Site Summary With Clarion 7

With a little time on his hands, Steve Parker sets about reorganizing his personal web site. A fairly straight forward C7 database application allows him to track his inventory, both published and yet-to-be-published, get reports and set his work schedule. And finally, it lets him create an RSS feed of new entries.

Posted Saturday, August 08, 2009

Writing Your Own Template Chain: Understanding Embeds

The Clarion template system is a toolset for building discrete pieces of an application and then combining those pieces in flexible ways while still allowing developers to add their own custom code. The embed system is the key to this flexibility.

Posted Monday, July 27, 2009

The Clarion Magazine Roadmap

These are interesting times for Clarion developers. For some, the biggest question how to get the most out of C6, and for others it's whether to migrate to C7. For yet another group the burning question is how to approach .NET development. How will Clarion Magazine respond to the changing development landscape? Read our roadmap and find out!

Posted Friday, July 17, 2009

Review: Clarion 7 Build 5768

Dave Harms takes a look at Clarion 7 build 5768.

Posted Wednesday, July 15, 2009

Clarion and .NET: The 50,000 Foot View

Clarion developers have long enjoyed the luxury of a single application framework (in two flavors). But how does the Clarion Win32 way of doing things compare to the received .NET approach(es)? Dave Harms takes a high level flight over the Clarion and .NET landscapes.

Posted Friday, July 10, 2009

- » Source Code Library 2009.05.31 Available
- » SoftVelocity Releases C7 Build 5646, C6 9059
- » A Brief History of Clarion (Updated)
- » C7 RC3 Released, Gold on April 13
- » Source Code Library 2009.03.31 Available
- » Call for Articles
- » Clarion 7 Release Candidate 2

[More free articles]

Clarion Sites

- » Andrew Popoff
- » Mitten Software
- » Clarion Jobs
- » DMOZ Clarion Third Party Page
- » French Clarion User Group
- » Clarion Australian Users Group
- » Atlanta Clarion User Group
- » South African Clarion User Group
- » NetTalk Central
- » Ingasoftplus

[More sites]

Clarion Blogs

- » John Griffiths
- » Build Automator Blog
- » Clarion Folk
- » ClarionX
- » AtlanWare
- » Johan van Zyl
- » RADBlog
- » Mark Sarson
- » DevDawn
- » Clarion Magazine

[More blogs]

Source Code Library 2009.06.30 Available

The Clarion Magazine Source Code Library has been updated to include the latest source. Source code subscribers can download the June 2009 update from the My ClarionMag page. If you're on Vista or Windows 7 please run Lindersoft's Clarion detection patch first.

Posted Wednesday, July 08, 2009

[Last 10 articles] [Last 25 articles] [All content]

Source Code

The ClarionMag Source Code Library

Clarion Magazine is more than just a great place to learn about Clarion development techniques, it's also home to a massive collection of Clarion source code. Clarion subscribers already know this, but now we've made it easier for subscribers and non-subscribers alike to find the code they need.

The Clarion Magazine Source Library is a single point download of all article source code, complete with an article cross-reference.

More info • Subscribe now

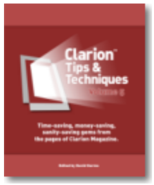
Printed Books & E-Books

E-Books

E-books are another great way to get the information you want from Clarion Magazine. Your time is valuable; with our e-books, you spend less time hunting down the information you need. We're constantly collecting the best Clarion Magazine articles by top developers into themed PDFs, so you'll always have a ready reference for your favorite Clarion development topics.

Printed Books

As handy as the Clarion Magazine web site is, sometimes you just want to read articles in print. We've collected some of the best ClarionMag articles into the following print books:



- » Clarion Tips & Techniques Volume 5 - ISBN 978-0-9784034-1-6
- » Clarion Tips & Techniques Volume 4 - ISBN 978-0-9784034-0-9
- » Clarion Tips & Techniques Volume 3 - ISBN: 0-9689553-9-8
- » Clarion 6 Tips & Techniques Volume 1 - ISBN: 0-9689553-8-X
- » Clarion 5.x Tips and Techniques, Volume 1 - ISBN: 0-9689553-5-5
- » Clarion 5.x Tips and Techniques, Volume 2 - ISBN: 0-9689553-6-3
- » Clarion Databases & SQL - ISBN: 0-9689553-3-9

We also publish Russ Eggen's widely-acclaimed Programming Objects in Clarion, an introduction to OOP and ABC.

From The Publisher

About Clarion Magazine

Clarion Magazine is your premier source for news about, and in-depth articles on Clarion software development. We publish articles by many of the leading developers in the Clarion community, covering subjects from everyday programming tasks to specialized techniques you won't learn anywhere else. Whether you're just getting started with Clarion, or are a seasoned veteran, Clarion Magazine has the information *you* need.

Subscriptions

While we do publish some free content, most Clarion Magazine articles are for subscribers only. Your subscription not only gets you premium content in the form of new articles, it also includes all the back issues. Our search engine lets you do simple or complex searches on both articles and news items. Subscribers can also post questions and comments directly to articles.

Satisfaction Guaranteed

For just pennies per day you can have this wealth of Clarion development information at your fingertips. Your Clarion magazine subscription will more than pay for itself - you have my personal guarantee.

Dave Harms

ISSN

Clarion Magazine's ISSN

Clarion Magazine's International Standard Serial Number (ISSN) is 1718-9942.

About ISSN

The ISSN is the standardized international code which allows the identification of any serial publication, including electronic serials, independently of its country of publication, of its language or alphabet, of its frequency, medium, etc.

Clarion Magazine

Clarion News

Search the news archive

ClarionLive Weekly Webinar #24 - Robert Paresi on iQ-Sync and Bruce Johnson on OddJob and WinEvent

The pre-Labor Day weekend ClarionLive Webinar features two new products, OddJob and iQ-Sync, plus the more mature WinEvent.

Posted Thursday, September 03, 2009

iQ-Sync

Robert Paresi's iQ-Sync is a little utility which simply copies and synchronize files. Simply setup an unlimited number of copy projects, and then when you need to execute it, highlight the project and press Run. iQ-Sync will notify you if files are in use, and can optionally wait until they are available. iQ-Sync is free.

Posted Thursday, September 03, 2009

LogFlash Half Price

The LogFlash audit trail and undelete templates are available at half price until September 8th. The cost during the sale is $97.50, with no annual maintenance charges. 60-day guarantee, demo available. To order use the promotion code "LogFlash 50% off".

Posted Thursday, September 03, 2009

Aussie DevCon Training Schedule

The schedule for the Aussie DevCon Training has now been determined, and includes Clarion.NET training by Pierre Tremblay from SoftVelocity. Due to the relative lateness of determining the training schedule the deadline for the Regular Early Bird Rate has been extended to September 13th.

Posted Thursday, September 03, 2009

Handy Zip'N Email and Handy Zip'N FTP

New versions of Handy Zip'N Email and Handy Zip'N FTP. Both apps are still unlimited. These new versions provide for up to nine configurations which can be recalled with a single click. The CHT query control used by both apps now includes autocomplete capability for easier use. Query keywords pop into place automatically after you type three or more matching characters. Both apps are pre-compiled versions of how-to-demos shipped in source .APP format with the CHT toolkit and can be built with either Clarion 6 or Clarion 7 and The Clarion Handy Tools.

Posted Thursday, September 03, 2009

Product Scope 7.9

Product Scope 7.9 is now available. Product Scope 7 provides a combination of tools for keeping track of comparative product information, places to buy, manufacturers, product images, located on the Internet and on local storage drives. Updated features include: Product Scope 7 Clipboard; Spreadsheet Named Output. 30 day trial available. Product Scope 7.9 is a major change from previous installs - previous Product Scope 7 users (7.5.337 or earlier) need to use the conversion program and follow the detailed instructions in the PDF, available at http://www.encouragersoftware.com/download.html. Product Scope 7.8.377 - use full install or web update method.

Posted Thursday, September 03, 2009

CommandBars Wrapper Template 2.00

Noyantis Software has released CommandBars Wrapper template version 2.00. This release is available in the members area in the web site; a new demo is also available. Modifications include: Codejock v13.1.0 compatibility added; 'Event Handler' facility added; 'Initial State' of controls can now be conditionally set; 'Initial State' of bars can now be set; 'State" of control can now continually mimic a standard clarion control; Alignment property added to control definition; Standard Windows Actions added (STD:Close, STD:Cut, STD:PrintSetup etc); Standard Clarion Actions now handled for bars that contain Clarion menu content type; Min / Max / Close buttons added for MDI Windows; More control types added to System Menu; Icons now automatically added to Project List (if filename prefixed with a "~"); Default Font attributes now amendable; Office 2007 Theme files can now be applied; Office 2007 Window Frame can now be applied; RibbonBar Tabs can now be hidden; Button styles can now be set per Bar in one command; 'Show Gripper' option added to bar definition; 'Show Expand / Customize' option added; 'Default Tooltip Style' option added; Extra Embed Points added; ImageRsc storage variable increased; BUG FIX: Save & Restore of Window locations corrected; BUG FIX: Height of bar area incorrect after bar minimized or moved (Application Frame Extension); BUG FIX: Popups were still being displayed when the control had lost focus. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Tuesday, August 25, 2009

EZChangeLog Reporter 1.7

EZChangeLog Professional 1.7 is now available. The main features of EZChangeLog Reporter add additional report and data query capabilities, including output to Microsoft Excel spreadsheets for EZChangeLog Professional data files. Version 1.7 adds custom reports. Use a simple, intuitive interface to quickly create and save an unlimited number of custom reports.

Posted Tuesday, August 25, 2009

CHT Build 13C1.00

CHT Build 13C1.00 has been released. Current CHT subscribers can start their Webupdaters (for C6 and C7) to bring their installations up to date with the latest code.

Posted Tuesday, August 25, 2009

EasyListView Version 1.00

An updated demo is available for EasyListView Version 1.00. Features include: New SQL connections and Clarion Database Drivers (tps) examples; Forecolor, Backcolor, Alternate row color (greenbar) selections; Images from BLOB / Image files; Example uses MDI; Added possibility for graphic data like ProgressBar and MultiImage; Added possibility to catch Double-Click and Right-Click on ListView. EasyListView is a Clarion wrapper around a .NET ListView. It makes the ListView easy to use in your Clarion applications and provides some neat extra functionality.

Posted Tuesday, August 25, 2009

Capesoft OddJob In Beta

From the folks that brought you easy access to Services comes easy access to Windows Job Objects. A Job Object allows groups of processes to be managed as a unit. OddJob allows you to start a process and "attach" both the Standard In, and Standard out from that process to your process. You can feed it the input it requires, either on the command line or via Standard In, and you can also capture the output via Standard Out. So there's no more black window, and there's no more having to guess the result. Because you can capture, and examine, Standard Out, you can be sure of exactly what happened. You also control the "Environment" that the other process has to run with. So you have complete control over the environment variables that the other process will "see" as it runs. OddJob also allows you to monitor the processes in a Job returning information like CPU and Memory usage. So if you attach a process to your job, you can then decide if the other process has consumed too many resources, and if necessary, kill it. And restart it. So if you want one program to monitor another, this makes an excellent way to do that. Better yet, if the process being monitored is something you wrote, then you could be more polite than just killing it, by sending it a message to shut-down or something. So another way to think of OddJob is as a process manager. OddJob was originally written to manage PHP scripts in conjunction with NetTalk 5. OddJob costs $79, but is currently on special for $59 while in beta.

Posted Tuesday, August 25, 2009

RADFusion International and Combit have severed ties

Combit and RFI are no longer doing business together. Russ Eggen will continue support on all code RFI has published.

Posted Tuesday, August 25, 2009

Clarion ProImage 2.1

Clarion ProImage 2.1 with live video image capture has been released. ProImage is a drop in Image Editor for Clarion programmers. It allows you to add full featured photo processing and image editing to your application in less than 10 minutes. It is available for Clarion 7.x, Clarion 6.x and Clarion 5.5G. Both ABC and Legacy templates are included. This is a free upgrade for anyone who purchased ProImage on January 1, 2008 or later. Developers who purchased before that date can upgrade for $59.95 USD. Note: Since version 2.0 of ProScan is also available, developers who own both products can save an additional $20 USD by upgrading both at the same time for $99.95 USD. The full version of ProImage is $249.95 USD and there is a discount for buying it as a bundle with ProScan.

Posted Tuesday, August 25, 2009

Comsoft7 / KwikSYSTEMS Anniversary 2-4-1 Sale

Comsoft7 and KwikSYSTEMS are offering a 2-4-1 (two for the price of one) limited time sale until the end of September. Purchase any product offered by Comsoft7 or KwikSYSTEMS(tm) ASSOCIATES and get any additional product of equal or lesser value as a free Anniversary bonus. Advertised Bundles are considered as a single product for these purposes.

Posted Tuesday, August 25, 2009

BoTpl 4.1 (Free)

BoTpl 4.1, a free template collection, is now available. Templates in this release support C4-C7x, some support ABC only, some support C55 and newer, some not available on C7. New in this release: Added BoAppLog template; Bug fixes.

Posted Tuesday, August 25, 2009

BSRef Application Analyzer Suite 1.0

Comsoft7 is releasing BSRef Ver 1.0 Application Analyzer Suite. This is a combination template and application to analyze your application. Shows where all files, fields, keys, globals, local variables are used and indicates which ones are not used or found. Shows all embeds, indicates orphaned embeds. Demo available.

Posted Tuesday, August 25, 2009

ClarionQuickBooksConnect Update

An updated version of ClarionQuickBooksConnect has been posted. This release fixes a GPF. Please use the same download link and install password.

Posted Tuesday, August 25, 2009

EasyListView Version 1.00 Coming Soon

EasyListView is a Clarion wrapper around a .NET ListView. It makes the ListView easy to use in your Clarion applications and provides some extra functionality. Features include: Automatically transforms data source (ver 1 supports SQL connections and XML files) into a fully functional ListView, including automatically sorting and grouping rows; Supports owner drawing, including rendering animated graphics and images stored in a database; Easily edit the cell values; Easily enable drag and drop; Supports 5 ListView views (SmallIcon, LargeIcon, List, Tile, Detail); Supports automatic grouping; Supports collapsible groups (Vista only); Columns can be fixed-width, have a minimum and/or maximum width, or be space-filling (Column Widths); Displays a "list is empty" message when the list is empty (obviously); Supports tooltips for cells and for headers; Supports checkboxes in any column as well as tri-state checkboxes; Supports alternate rows background colors.; Supports custom formatting of rows; Supports searching (by typing) on any column; Supports image and text overlays. Requirements: Clarion 6.3 / Clarion 7, ABC or Legacy Template Chains, 32-bits only, .NET Framework 2.0 and above, Windows XP and above. Demo available. Price: $95 (New Subscription: 1 Developer license + 1-year Maintenance Plan). Release date: Oct 01, 2009 (will be shipped in October 2009). Limited time offer: $75 for all buyers who will purchase before the release date.

Posted Tuesday, August 25, 2009

SetupBuilder 7.0 Installation System

Lindersoft has announced the SetupBuilder 7.0 Installation System, the first installation system for Windows 7 (RTM) and Windows Server 2008 R2 (RTM). Any organization planning to move its applications to the Windows 7 platform will be able to use SetupBuilder to comply with this new industry standard. Version 7.0 includes new and enhanced functionality that will further reduce the time needed by software developers and systems administrators to create installations and distribute them to end user desktops in any Windows environment. It provides updates to the easy-to-use Visualizers that includes Lindersoft's point-and-click method of scripting technology for simple creation of installations. Version 7.0 also includes enhancements to the Script Editor, which developers use to create powerful, customized installations. Available in Developer and Professional editions. SetupBuilder starts at $199.00 for a royalty-free usage license. A trial version is available.

Posted Tuesday, August 25, 2009

CommandBars Wrapper Template Demo

A new demo app has been uploaded to show off some of the new features added to version 1.19 of the CommandBars Wrapper template.

Posted Wednesday, August 05, 2009

Noyantis Office Closure

Andy Wilton be away from Thursday 6th August until Monday 17th August. During this period he will be collecting emails periodically but replies may be delayed for several days.

Posted Wednesday, August 05, 2009

SkinFramework Wrapper Template 1.03

Version 1.03 released of the SkinFramework Wrapper template is now available. Changes include: Codejock v13.1.0 compatibility added; Exclude Modules facility added; New EnableDialogTexture method; New ExcludeModule method. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, August 05, 2009

TaskPanel Wrapper Template 1.10

Version 1.10 of the TaskPanel Wrapper template is now available. Changes include: Codejock v13.1.0 compatibility added; Event handler facility added; Item content type added to allow for easier integration with other Noyantis templates; New SetCtrlIcon Method; New SetGroupIcon method; Extra embed points; ImageRsc storage variable increased. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, August 05, 2009

PropertyGrid Wrapper Template 1.09

Version 1.09 released of the PropertyGrid Wrapper template is now available. Changes include: Codejock v13.1.0 compatibility added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, August 05, 2009

DockingPane Wrapper Template 1.04

Version 1.04 of the DockingPane Wrapper template is now available. Changes include: Codejock v13.1.0 compatibility; ImageRsc storage variable increased. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, August 05, 2009

ShortcutBar Wrapper Template 1.19

Version 1.19 of the ShortcutBar Wrapper template is now available. Changes include: Codejock v13.1.0 compatibility added; Extra embed points added; Item content type enhanced to allow for easier integration with other Noyantis templates; Navigation Tree visual selection error fixed in C7.

Posted Wednesday, August 05, 2009

Thin@ 1.21

Thin@ 1.21 is now available. Changes include: Updated server side components with new distributed services; Updated server side application with new compiled applications; Updated server application folders with new library file distributed with this installation; Legacy template support; Improvements to listbox support; Menu colors, styles and fonts supported; Prop:Angle supported; Window resizing; %Username and %Password variables; Various session variables available in server side application; Speed improvements and bug fixes.

Posted Wednesday, August 05, 2009

TCD 2.5.03

TCD 2.5.03 is now available. Changes to the version control system include: New function to add a full folder (and its sub-folders) in the window of projects; New option to delete a set of files; New option for rename a computer; New option for project labels; Fixed a problem related to refresh after some actions (like Get Last Version, for example); Change when TDC minimizes on systray; Set working folder enhancement; Various UI improvements and fixes. Changes to the tracking system include: Added a summary for each Project in the main window with different kinds of totals by issue for the selected Project; A new option in the main menu to maintain the categories and subcategories for incidents; The filters area was modified, rearranged controls, and in the toolbar there are new filters that are activated based on the selected column; Added new controls in the procedure for sending emails; Add, modify and delete incidents; Maintain categories and subcategories for incidents; Improved speed; User interface improvements and more.

Posted Wednesday, August 05, 2009

J-Media 2.0

J-Media 2.0 is now available.

Posted Wednesday, August 05, 2009

CoolFrames 1.21 Beta

CoolFrames 1.21 beta is now available. Please note that the install file requires an unlocking code which is supplied when you purchase CoolFrames. Changes include: Added support for right to left (RTL) windows layout; Fixed bug where after ALT key use menu's become unresponsive; Fixed bug where MDI Child close / restore buttons not correctly redrawn; Fixed bug where in DLL mode ALT press resulted in GPF; Fixed bug where MDI Child background image is trashed on restore from maximized; Fixed bug where window with status bar changed size; Fixed GDI handle and memory leak.

Posted Wednesday, August 05, 2009

Clarion Magazine

The Clarion Magazine Roadmap

by Dave Harms

Published 2009-07-17

These are interesting times for Clarion developers. Recently I published an article titled Clarion and .NET: The 50,000 Foot View. In that article I discussed some of the significant differences between Clarion Win32 development and .NET development. And as Bruce Johnson pointed out in a comment to that article, there really is a divide in the community over how (or even why) Clarion should "do" .NET.

A lot (possibly a majority) of Clarion devs simply aren't concerned about .NET and neither are their customers, who just want software that works.

At the other extreme are those Clarion devs who have a pressing need for .NET but lack a Clarion-like path through the .NET landscape.

A focus on Win32

Here at the mag we've had a lot of discussions about how to best deal with these diverging paths. And what it comes down to is this: Win32 is still the sweet spot for most Clarion developers. Until that changes, Clarion Magazine will remain primarily a publication about Win32 development. We'll continue to cover important Clarion# news, of course. And once there's a .NET AppGen and at least one Clarion# template chain you can expect an increase in .NET coverage. But the reality is that most Clarion developers are far more concerned about the migration from C6 to C7 (or just getting the job done in C6) than they are with moving to the .NET framework.

If you are one of those devs looking to .NET, take heart. We have something special in the works for you, which I'll get to in a moment.

Clarion Magazine v.3

To better support Clarion developers in their Win32 work we've been working on a major update to the ClarionMag site. This is actually the third major version of the mag. The very first version was, if you can believe it, a static site. It had a flat file of logins for authentication and authorization, but other than that there was no back end database; each web page was a discreet HTML page. ClarionMag quickly outgrew that crude architecture, and for most of the last decade we've delivered the magazine using a combination of SQL, Clarion, and custom server-side code. We're now taking that server-side code to a new level with Microsoft's ASP.NET MVC framework.

ClarionMag version 1 was just a collection of web pages; ClarionMag version 2 went beyond pure article content and added reader comments, search functionality, a topical index, RSS feeds, surveys and a blog.

Here are just some of the things we're adding to ClarionMag version 3:

- Improved site navigation and search
- File uploads - share your Clarion code
- A third party product directory
- User ratings (of third party products and other site elements)
- Skill sets - let other Clarion developers know what you do well
- Favorites list - keep track of your favorite articles/links/products and optionally share that list with others

Our vision for ClarionMag version 3 is as much about online community as it is about publishing. We want the mag to be a place not just where you read content, but where you can share information with other developers.

We've enjoyed and appreciated the support of the Clarion community for the last ten years; we trust that Clarion Magazine version 3 will be a valued resource for many years to come.

A new .NET magazine

Along with a new ClarionMag site we'll be rolling out a new online magazine for .NET business software developers. Although we think this new mag will be immensely useful for any Clarion developer doing .NET work, it really isn't about Clarion per se. It's about .NET, and its purpose is to help business software developers choose the optimum path for their needs path among the many different paths available.

Our new .NET mag will have many of the features of Clarion Magazine, but with a .NET focus. These features include:

- Articles focused on business software development
- News items
- Site navigation that actually helps you plan and execute your .NET strategy
- Rated links to the best .NET business programming resources
- Favorites list - keep track of your favorite articles/links/products and optionally share that list with others
- Skill sets - let other .NET developers know what you do well
- Glossary of terms
- And much more

We're very excited about the new .NET mag and the changes coming to ClarionMag. These two publications will offer terrific new opportunities to learn, to share your knowledge, and to make the most out of your business software development options.

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors (ASJA).

Reader Comments

Posted on Monday, July 20, 2009 by Kevin Dohren

Hi David,

As you say exciting times - lots of questions on how this will work, for starters...

Will you need a separate subscription to access it?

Will existing Clarion.NET and Clarion# articles be moved over to it, and only be accessible there?

Kevin

Posted on Monday, July 20, 2009 by Dave Harms

Kevin,

Yes, the new site will be a subscription-based publication and will be a separate entity from Clarion Magazine. There will probably be a discount for those who want both, but I haven't worked out the details yet.

I will not be moving any existing Clarion# or Clarion.NET articles to the new site. And while the new site might occasionally carry some Clarion-specific content, that won't be its primary focus.

So will the new site be of interest to existing Clarion developers? If they're doing any .NET work, or have any interest in doing .NET work, I think the answer is an emphatic Yes!

One of the great features of Clarion development is that it provides a standard architecture for application development. This, I think, is the foundation of the Clarion community: we have a common language and a common basis for discussion.

By comparison, the .NET world presents a bewildering number of architectures and development paths and a vast universe of tools and add-on products. This fragmentation makes it harder to form a cohesive view of what your particular kind of .NET development should look like, and hard to find the help you need.

Template sets from SoftVelocity (or third party vendors) will help, but only to the degree that the choices made by the template designers align with your specific requirements. Template sets are unlikely to cover such issues as multi-language development, continuous integration, test driven development, domain-specific languages, or any of dozens of other significant concepts common in .NET development.

We believe we've found a way to bring order out of that chaos, to impose some structure on the massive amounts of .NET programming information out there, and to make it easier for developers to find their way through the .NET universe whether they're using Clarion or any other toolset. And that's the purpose of the new site.

Dave

Posted on Monday, July 20, 2009 by Dave Harms

Kevin,

Yes, the new site will be a subscription-based publication and will be a separate entity from Clarion Magazine. There will probably be a discount for those who want both, but I haven't worked out the details yet.

I will not be moving any existing Clarion# or Clarion.NET articles to the new site. And while the new site might occasionally carry some Clarion-specific content, that won't be its primary focus.

So will the new site be of interest to existing Clarion developers? If they're doing any .NET work, or have any interest in doing .NET work, I think the answer is an emphatic Yes!

One of the great features of Clarion development is that it provides a standard architecture for application development. This, I think, is the foundation of the Clarion community: we have a common language and a common basis for discussion.

By comparison, the .NET world presents a bewildering number of architectures and development paths and a vast universe of tools and add-on products. This fragmentation makes it harder to form a cohesive view of what your particular kind of .NET development should look like, and hard to

find the help you need.

Template sets from SoftVelocity (or third party vendors) will help, but only to the degree that the choices made by the template designers align with your specific requirements. Template sets are unlikely to cover such issues as multi-language development, continuous integration, test driven development, domain-specific languages, or any of dozens of other significant concepts common in .NET development.

We believe we've found a way to bring order out of that chaos, to impose some structure on the massive amounts of .NET programming information out there, and to make it easier for developers to find their way through the .NET universe whether they're using Clarion or any other toolset. And that's the purpose of the new site.

Dave

Posted on Saturday, July 25, 2009 by Lee White

Sounds like a excellent endeavor and if you include something akin to Clarion 101, such as dotNet 101, maybe I'll eventually grasp the concept and consider making an effort!<g>

Posted on Tuesday, August 04, 2009 by Dave Harms

Lee,

Thanks, and yes, I think this will definitely enable a "101" kind of overview of the available information.

Dave

Add a comment

Clarion Magazine

Topspeed Superfiles

by Steven Parker

Published 2009-08-28

TopSpeed superfiles, which allow you to combine multiple TPS "files" into a single physical file, have a curious reputation. Many Clarion developers use them (I use them). Many who do use TPS files categorically refuse to use superfiles claiming that they are unreliable.

TPS files were first introduced in Clarion 3 (Clarion Database Developer, CDD, sometimes known as Clarion for DOS, CFD). A little known fact is that the TPS driver was originally created as a proof of concept for the Driver Kit sold by Clarion Software for a short while.

The TPS file system is modeled on Btrieve files, believed, at the time, to be one of the best file systems for PC systems (I am not a subscriber to that belief, by the way). In fact, Clarion 3 APP and DCT files are Btrieve files (if you've been upgrading your Clarion installation for some years, this may explain why you sometimes get an error on Btrieve not being loaded; Btrieve was last distributed with Clarion 5.0).

An important feature of both file systems is that keys are stored inside the data file. This means that TPS and Btrieve DAT files do not use up file handles the way Clarion DAT, dBase, Clipper, FoxPro, etc., files do. These file systems use a handle for the data file and a handle for each key and memo.

TPS files use one handle for the data file and (all) keys and all memos. In the mid 1990's, this was an important issue. Because CDD and (pre-ABC) Clarion for Windows did not support methods like SetOrder and AppendOrder, Clarion developers were making keys for each browse order and each report order.

Keys were multiplying and handles were being exhausted, especially on networks, resulting in runtime errors and program failure. Dynamic Indices were introduced in CDD, with some very handy options when used with TPS files (like filtered builds), in my opinion eliminating the need for report-specific keys. Still, keys multiplied and converting Clarion DAT files to TPS was not a trivial undertaking (FM2 had not yet been invented).

Besides combining data, keys and memos into a single file, there were two significant enhancements TPS offered over Btrieve.

1. TPS files are runtime compressed to save disk space
2. TPS files can store multiple tables in a single physical file further economizing on handles

Runtime Compression: TPS files have the peculiar feature that they sometimes get larger on deletes and smaller on inserts (during the writing of this article, I noticed that MS Word DOC files can do the same thing). Usually, however, TPS files grow less than the cumulative sum of the record's sizes. This is due to the files (mainly the string data) being compressed at runtime. As the file driver documentation states:

> The TopSpeed driver compresses the entire record buffer area (not individual fields within the record), therefore, compression gains can be realized by placing similar fields adjacent to each other in the FILE declaration.
>
> The TopSpeed file system uses the same compression algorithm for STRINGs and MEMOs; however, the compression occurs at a "higher level" for MEMOs than for STRINGs. As a result, MEMOs do have a disk space advantage over large STRINGs

> (over 500 bytes) and smaller STRINGS can have a slight performance advantage over MEMOs. The larger the STRING, the greater the advantage.
>
> MEMOs do carry the advantage of BINARY versus NONBINARY, plus MEMOs may be omitted from all processing with the NOMEMO statement.
>
> STRINGs are always allocated space (RAM) within the record buffer, whereas MEMOs are only allocated space when the file is OPENed. Also MEMOs cannot be key components.

This is another element in the driver's design to reduce system resource requirements and increase performance. Moving a compressed file across a network (remember, with "flat" files, the entire file moves) will be more efficient than moving an uncompressed one.

Multiple Tables: This is the feature which gives rise to the term "superfile." Again, the documentation states:

> The TopSpeed driver can optionally store multiple tables in a single file. This lets you open as many data tables, keys, and indexes as necessary using a single DOS file handle. This feature may be especially useful when there are a large number of small tables, or when a group of related files are normally accessed together. All keys, indexes, and memos are stored internally.

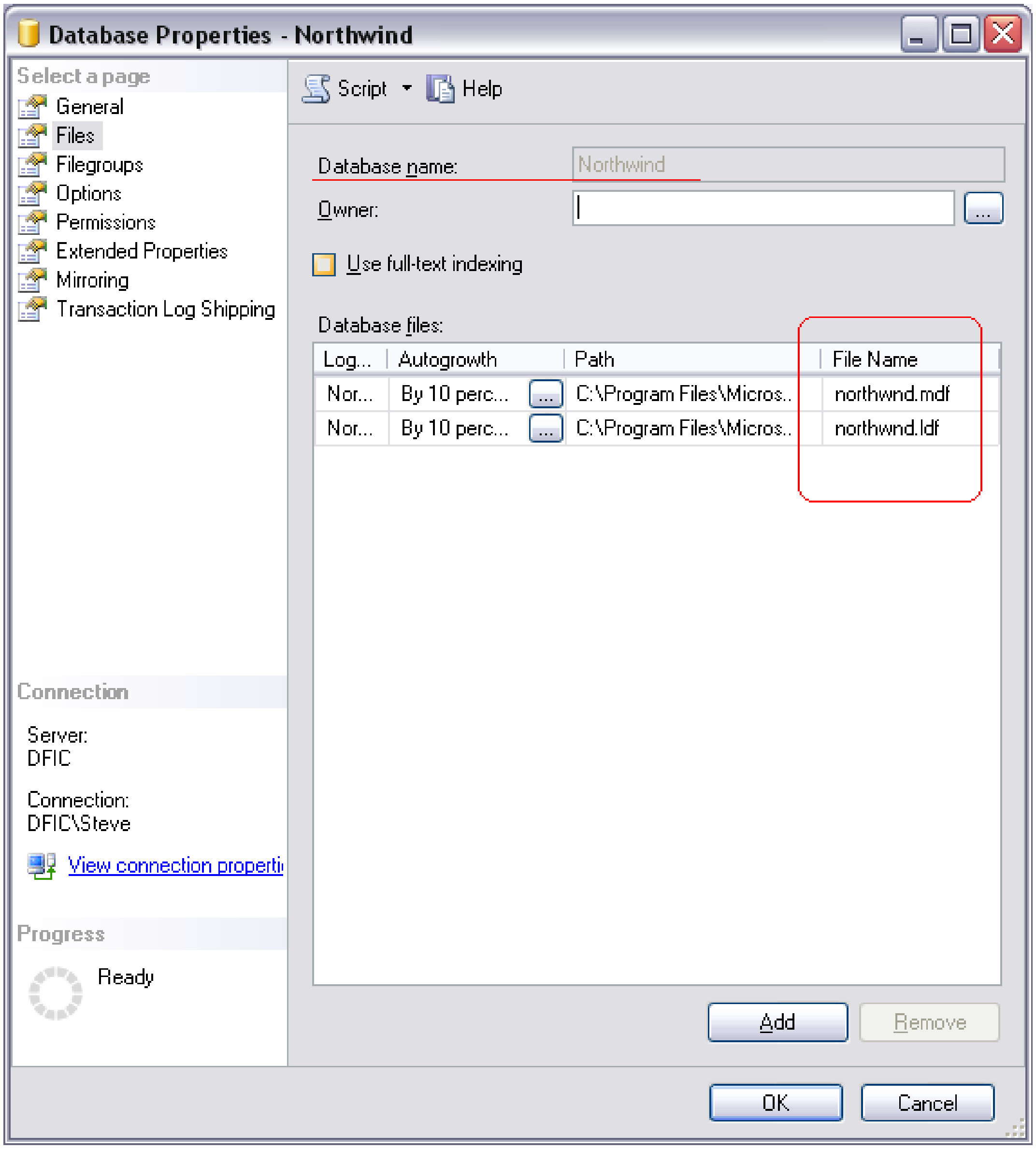
Theoretically, all the data, keys and memos in an entire application could be opened using only a single file handle.

There are two terms here that are not normally used together: *file* and *table*. *File* is a concept we all understand. So is *table*. The term *file* is (usually) used to refer to a flat file, like a Clarion DAT file or a dBase DBF. A *table* is typically a SQL entity, usually a logical entity and not a physical entity.

When talking about TPS superfiles it becomes very important to use the correct meanings of these terms. In superfile terms, a file is an entry in the O/S' file system, a physical entity on disk. It is what you see in Windows Explorer's or Total Commander's file listing, for example. A table is a related group of records or rows.

This is the way Microsoft SQL is organized. MS SQL has files. A single physical file on disk, for example Northwind. MDF, contains multiple tables, logical entities, not one of which is called "Northwind."

One physical file (Figure 1) can contain multiple tables, or logical files (Figure 2):



Database name: Northwind

Owner:

□ Use full-text indexing

Database files:

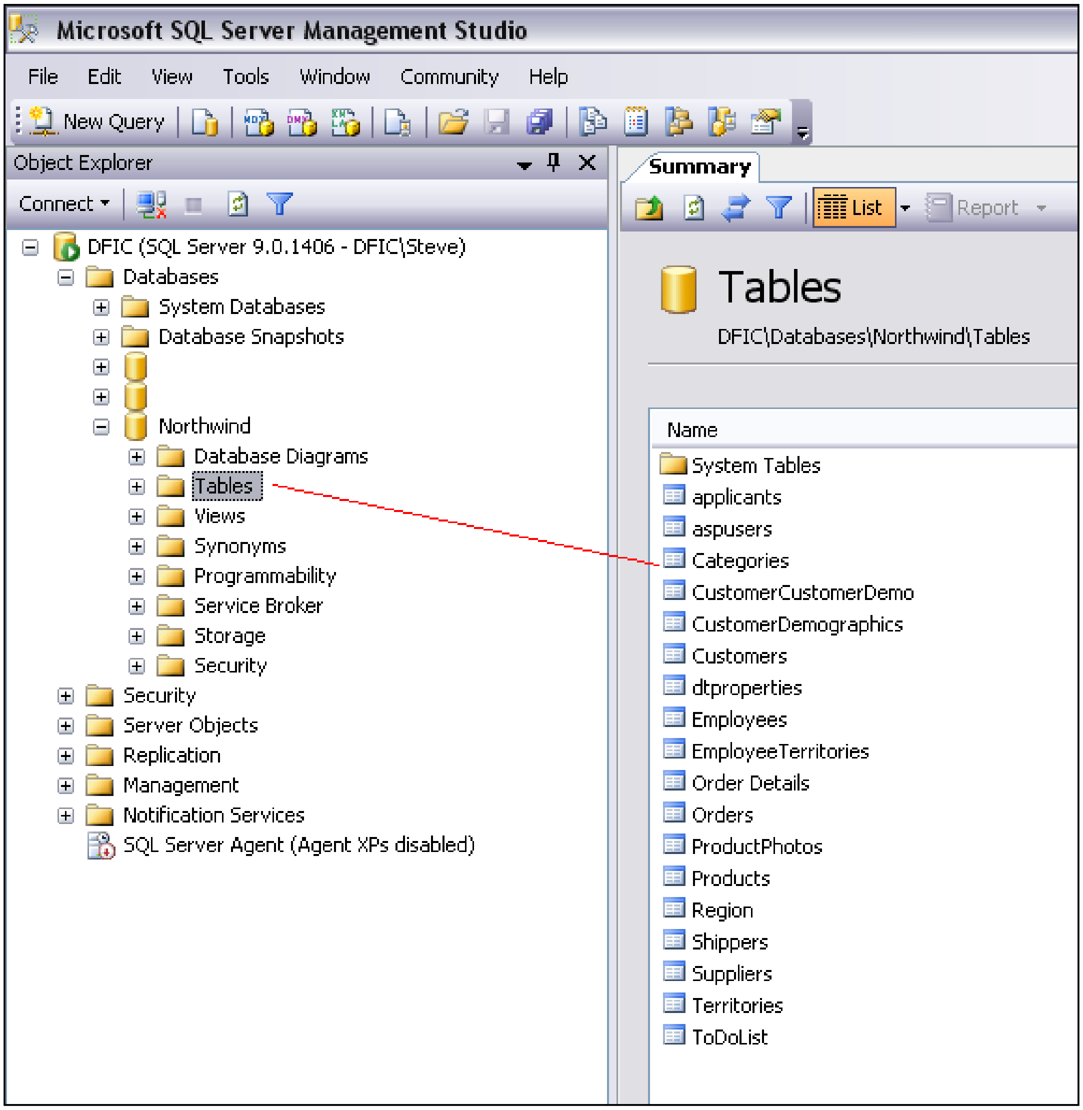
Log...	Autogrowth	Path	File Name
Nor...	By 10 perc...	C:\Program Files\Micros..	northwnd.mdf
Nor...	By 10 perc...	C:\Program Files\Micros..	northwnd.ldf

Server:
DFIC

Connection:
DFIC\Steve

Ready

Figure 1. MS SQL specification of its files



```
DFIC (SQL Server 9.0.1406 - DFIC\Steve)
  Databases
    System Databases
    Database Snapshots


    Northwind
      Database Diagrams
      Tables
      Views
      Synonyms
      Programmability
      Service Broker
      Storage
      Security
  Security
  Server Objects
  Replication
  Management
  Notification Services
  SQL Server Agent (Agent XPs disabled)
```

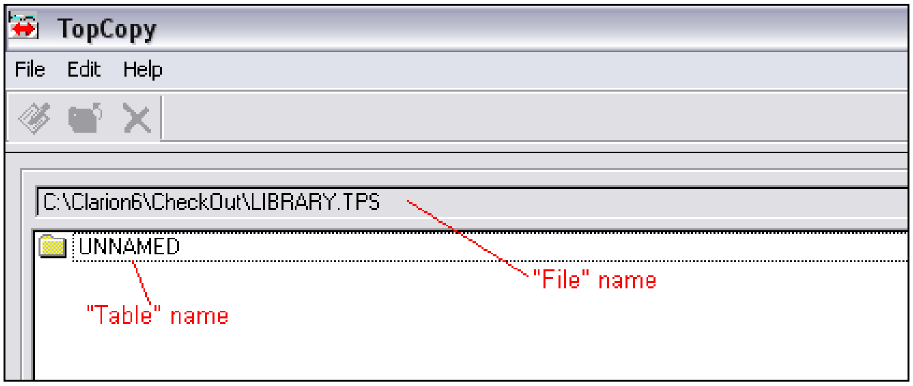
```
System Tables
applicants
aspusers
Categories
CustomerCustomerDemo
CustomerDemographics
Customers
dtproperties
Employees
EmployeeTerritories
Order Details
Orders
ProductPhotos
Products
Region
Shippers
Suppliers
Territories
ToDoList
```

Figure 2. Tables within a single file

Therefore, TPS superfiles are structured just like some SQL databases (N.B.: not all SQL engines store tables inside a single physical file, some store one table per physical file, some store indexes separately, some split tables across multiple files, etc).

In fact, single table TPS files (that is, TPS files not containing multiple tables) are also structured this way. Under certain circumstances, usually unhappy ones (for me it is usually a file system error), you will find that there is a default table name, "unnamed," in single-table TPS files.

One place anyone can see this "unnamed" table, at will, thus proving that all TPS files are actually superfiles, is when opening a TPS file in TopCopy:



C:\Clarion6\CheckOut\LIBRARY.TPS

UNNAMED

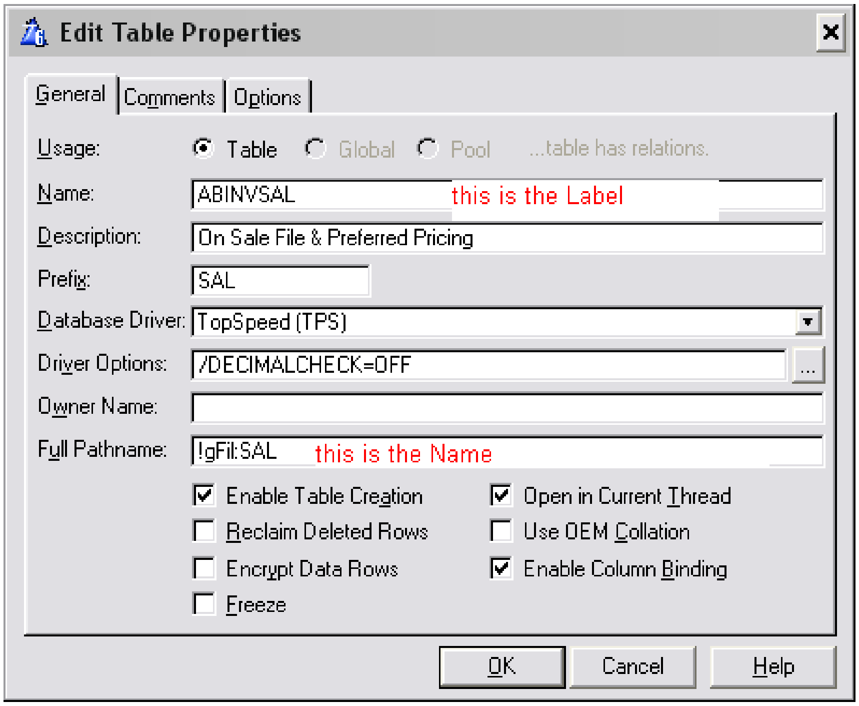
Figure 3. Unnamed table visible in TopCopy

So, it turns out to be advisable to think about TPS files, and to talk about them, in the same way one thinks and talks about SQL files.

There is a third, Clarion-specific, concept I need to remind you of: *label*. When I create a file or table in the Dictionary Editor or in a data embed, I specify a label. I do not specify a file name, nor a table name, I specify a label.

A label is what I use in my code (technically, a label identifies a place in the code). If a file has the Name attribute, the Name is how the operating system refers to the entity. In the absence of a Name, the label and the Name will be the same (Clarion plugs the label into the Name for me - note that when using SQL, should you let Clarion use the label for the Name, it will truncate it to eight characters). But my code never uses the Name, only (and always) the label.

Consider:



Usage: ● Table ○ Global ○ Pool ...table has relations.

Name: ABINVSAL

Description: On Sale File & Preferred Pricing

Prefix: SAL

Database Driver: TopSpeed (TPS)

Driver Options: /DECIMALCHECK=OFF

Owner Name:

Full Pathname: !gFil:SAL

☑ Enable Table Creation ☑ Open in Current Thread

☐ Reclaim Deleted Rows ☐ Use OEM Collation

☐ Encrypt Data Rows ☑ Enable Column Binding

☐ Freeze

Figure 4. Dictionary file declaration with name

The label in Figure 4 is ABINVSAL. In my code, I open, close, set, add, put, etc., ABINVSAL.

However, on disk, there is no ABINVSAL.TPS. In this case, depending on the value I assign to gFil:SAL, it could be one of several files: SALEFILE.TPS, PREF1.TPS, PREF2.TPS, etc.

The problem with superfiles

In reality, the perceived reliability problem with superfiles isn't superfile-specific, as Figure 3 proves. The problem is TPS-specific ("TPS-wide"?).

TPS files got a lot of negative comment when Clarion for Windows apps started to go into production. It took the better part of two years for everyone to become convinced that the majority of these early problems turned out to be hardware problems. Bad NICs were the number one cause of corrupted TPS files. (In fact, I remember at one point, if someone complained of corruptions and turned out to be using a specific brand of network card, people just stopped responding, so bad was that one brand.) Substandard wiring was #2.

> We've all heard about the customer who strung his network cable over the ballast in his fluorescent lights. How about the one with Cat5 between PCs and the wall but Cat3 inside the wall?

TPS corruptions went away once decent hardware was implemented. However, that, in a sense, begs the question - Why should TopSpeed files be more sensitive to hardware than other file systems?

In the on line help, press the Database Drivers button. Then select the Topspeed Database Driver and find the link for "Repairing Data Corruption in TopSpeed/Clarion Files." There's a wealth of important information in this article.

Specifically,

> TopSpeed files are not more susceptible to corruption than other files; however, error detection is higher because the driver engine is extremely sensitive to the smallest anomaly in the data file.

During runtime compression, the driver checks the file. When accessing the file, the driver also checks the file. If the "old" and "new" computations do not match & error!

> If a bad bit is transferred and then written into the file, it is usually identified, by the driver, when the record is decompressed. During decompression, the actual record size is compared to the size reported in the header. If either of these was written incorrectly to the file, the two sizes will not match. This detection will cause the driver to report an error or corruption in the file.

What is different here is that TPS files are checked regularly. Clarion and Btrieve and other flat files are not. A flat file can have an error in it that goes undetected for years.

I have seen Btrieve files which had a field width increased by one character. The error was not detected until a new version of the software started spitting out errors (and not ErrorCode 47, which might have helped). I have seen FoxPro and Btrieve files "inherit" high order ASCII characters in unexpected places and the user doesn't know until a report shows bizarre subtotals.

I have seen an error in a Btrieve file go undetected for 10 years. And the actual error file didn't report a problem, a related file did. It took three days to track down the cause of the error.

So why is so much special attention paid to superfiles? Simple. When you corrupt a single-table TPS file, at most you lose only a single table. When you corrupt a multi-table TPS file, you could lose everything. This is not unlike what happens if an MS SQL file becomes corrupt (somewhat more likely to be through a hardware error, like a flaky hard drive, but it does happen).

So, which is better? Error now? Error in the future? The purist in me says "let me know ASAP so I can fix it." The pragmatist in me says "the user may never run into problems." Of course, if and when that user final does hit the problem, it's going to be a big one (the voice of experience speaks). There simply is no "right" answer to this dilemma and, perhaps, that's what is actually so frustrating about superfiles.

Uses for superfiles

The documentation suggests two candidates for superfiles:

- files that are usually accessed together
- numerous small files

Files that are usually accessed together? Think "parent-child." On reflection, it seemed to me that these could also turn out to be my largest and most frequently updated files. Given the possibility of a "total loss," I am loath to use them for this purpose simply because a group of files constitute a set.

"Small files?" Yes! I have lots of little validation files (look up files). I find this a perfect use for superfiles. I am the only one who updates these files and maintaining them all inside a single superfile reduces directory clutter. In the POS system I used to work on, a single superfile holds item types, pricing rules, supported tag layouts and supported bar code symbologies, among other things. All of the tables I've just listed are look ups from the master inventory file.

My Clarion internet apps all use superfiles exclusively. My knowledge base, my bibliography of Clarion journals, my download center each uses a single superfile to service their data needs. Using superfiles reduced handles, traffic, bandwidth requirements and, most important, the number of files I have to remember to upload when updating my sites. (More accurately, superfiles reduce the number of files I can forget.) My sites have been running for years without issue (or, at least, without issue causes by the files).

Next time: superfiles and Name().

Steve Parker started his professional life as a Philosopher but now tries to imitate a Clarion developer. He has been attempting to subdue Clarion since version 2007 (DOS, that is). He reports that, so far, Clarion is winning. Steve has been writing about Clarion since 1993.

Reader Comments

Add a comment

Clarion Magazine

The C7 Experience

by Steven Parker

Published 2009-08-11

In my previous article I described a C7 application I use to create an RSS feed for a new page on my web site. Apart from the tutorials (which I still strongly encourage everyone to do completely) the RSSBuilder app is my first new development in C7. This seems appropriate; after all, it's not exactly classified information that I'm a C7 proponent.

What was my experience like?

Over all? I still have a huge learning curve. Most of the pieces are there. The edges are still rough.

RSSBuilder.APP is built with 5768. When I installed 5768 I did not do a clean install. I installed over the previous release. Therefore, some of the less than pleasant things that happened may well be caused by that (why this should have to be a caveat is an issue in itself but not one I want to address here).

From previous articles, I don't suppose it comes as a surprise to CMag readers that I am particularly enthusiastic about the meaning C7 gives to “integrated” in “Integrated Development Environment” (IDE). C7 gives me a single environment for all my Clarion work in all the versions I have installed and use for development.

In those articles, I have shown how to set up C7 to maintain Clarion 6 applications as C6 apps. I have also moved some 5.5 apps successfully to C7; these are demo apps from previous CMag articles. I haven't done new development in 5.5 for years and keep it around only for CMag demo apps and for maintaining my Internet Connect and Web Builder apps. Once I can use C7 to maintain my IC and WB apps, I intend to stop using C6 also.

Mask

The single biggest thing holding me back from moving aggressively into Clarion 7 is the broken MASK attribute. This is, perhaps, disproportionately important to me. But the fact is that date fields appear everywhere in my work. I use D1 pictures and Insert mode for my dates and C7 mashes dates unless the user enters a leading zero. I have found no workaround.

Note to self: I have some good news and some bad news. The bad news is that MASK is still broken for date fields @D1. The good new is that, without MASK, date fields using @D17, populated by a form wizard, work as expected when the user enters mm/dd/yy (without requiring a leading zero). However, trying to retro-fit this to a converted C6 app failed.

The IDE

Now, it's a little known fact, that although I'm a proponent of C7, somewhat paradoxically, I don't much care for the new IDE. I also didn't much care for the Clarion 3 IDE. Actually, I thoroughly hated the C3 IDE at first. At one point, while rewriting a CPD app in C3, I was literally in tears.

I got used to the Clarion 3 IDE, learning to love it. Indeed, within a few months I arrived at the point where I had the IDE jumping through hoops for me. I expect that history will repeat itself and I'll get used to the Clarion 7 IDE also and look back on the C6 IDE as totally archaic.

Dictionary Editor

Speaking of Clarion 3, the new dictionary editor seems to share a feature with that older IDE. In Clarion 7, I am prompted to save changes when I move to a new item, file or field.

I know that there are a number of folks out there who don't care for this. Indeed, I was one of them, even though I've seen this before. Note my use of the past tense. In creating the application for this article, I had the … uh, opportunity to change both file layouts and key definitions. On one occasion, adding an attribute to a key, I did not get the save confirmation. I didn't notice. The change didn't take. Maybe this isn't such a bad feature.

Pads

I have my IDE configured so that Solution and Properties pads are pinned. All other pads are unpinned. The Control Template pad constantly intrudes. The Control Template Pad expands whenever I open a procedure. Clicking on something retracts the pad and moves the Properties Pad into place. I find this inconvenient but I can live with it, for the time being. Yes, it's a bug but it's not a show stopper.

Reports

When I dragged variables from the Data Pad onto the report, they dropped on the upper left of the target band. As I recall, this happened mostly when dragging local variables into the page header.

This is a serious problem, not a show stopper, but very serious.

Populating lines was … an adventure. I populated a line. It looked okay in the Report Formatter. It looked okay in Report Preview. But when I compiled and ran the report, the line was at a steep angle.

Examination of the code showed that the fourth parameter of LINE was omitted. Entering a zero (much easier than trying to re-drag the line's end point to the proper position) fixed the problem. I consider this a moderately serious problem.

Much more serious was an unpredictable dropping of prefixes. There are three reports in RSSBuilder.APP and I had this happen in each one (and, I think, on a Form or two). After dragging variables into the detail band and compiling, I got “unknown variable” errors.

Clicking the error and examining the source showed that the prefix wasn't there. Yes, the ability to edit the source made this an easy fix. This is not a show stopper but it is as close to being one as you can get.

Window preview

Window preview still hasn't arrived and I really miss it.

I created RSSBuilder.APP as an SDI app. This is the first time I have ever created an SDI app and it is a mistake I'm not likely to repeat in the near future.

That error aside, when I populated a background image, I had to compile and run to see what it looked like. Because I didn't like the first several images I had chosen, this ended taking quite a bit of time. Unnecessarily, in my opinion.

Similarly, I was unable to see that Extended UI settings weren't working. Was this because the app was SDI, not MDI? The documentation does not limit the Extended UI to MDI frames.

Now, the C7 compiler is very quick, especially in Conditional Generation mode, so this could have been much more painful. Still, Preview is a long-time Clarion feature and I look forward to its reinstatement.

> **Editor's note:** SoftVelocity has announced there will be a new window previewer in Clarion 7.1, and has posted a blog entry

with screen shots.

Wizards

I usually start Browses and Forms with a Wizard. This allows me to begin with a degree of consistency in button sizes and icons.

Changes to the Wizards were not being saved. It was fixed. It's broken again. PITA.

Focus Loss

The IDE sometimes seems to lose focus. Returning to the app tree after working on a procedure requires a double click on the Application Pad to generate, build or build and run. Save and close an app and a double click on the “X” is required to close Clarion 7.

I'm developing a habit of clicking on a blank area in the Application Pad immediately on returning to the app tree to refocus the IDE.

Duplicate declarations

Declaring LOC:Date in Local Data in multiple procedures in the same module caused duplicate declaration errors. Moving procedures to different modules “cured” it. I have had this happen, if I recall correctly, in earlier versions of Clarion, so this may not be new to C7.

This is moderately serious, especially because, unlike C6, these errors keep coming up on subsequent compiles.

The verdict

It still looks like I cannot move my Clarion 6 apps under the C7 umbrella because of issues with MASK and date entry. And until I move my C6 apps into C7, I'm not going to try my IC / WB 5.5 apps. Other than that, I found no show stoppers.

I did find a number of problem items. Individually, they are small. Taken together (and you must take into account any effect of not having done a clean install), they constitute a disappointment. I am so ready for the move forward (did I mention how fast the IDE starts and runs in Vista 64?) Despite that, I think C7 is getting closer to becoming my Clarion development environment, not as quickly as perhaps I would have liked, but it is getting there.

Steve Parker started his professional life as a Philosopher but now tries to imitate a Clarion developer. He has been attempting to subdue Clarion since version 2007 (DOS, that is). He reports that, so far, Clarion is winning. Steve has been writing about Clarion since 1993.

Reader Comments

Posted on Wednesday, August 12, 2009 by mogamat abrahams

Thanks for the review, if you purchased a csp for 6.3 (with the promise of 7), do you consider the terms of that contract fulfilled by the 5768 build?

Posted on Wednesday, August 12, 2009 by Steven Parker

Your question is a bit like asking a VB developer how he feels after trying to maintain his apps in a new version of VB (given MS' infamous failure to provide backward compatiblity).

I can do new development and get the results I expect, so in that regard, yes, 5768 fulfills the bargin. In that I cannot yet replace C6, no. (OTH, this means that I don't yet _have_ to implement my plan to buy a new PC with a 64 bit O/S just yet. When C7 can maintain all my previous apps for me, I plan to replace my XP development machine. <g>)

It is important to keep in mind that each of us has differing needs. I was able to go into live production with CDD3004, a release many considered completely unuseable. What's the expression?-"You mileage may vary"?

For example, MASK is a show stopper for me. But I know several Clarion developers who weren't even aware it was broken. Several NG posting in the last few weeks indicate that there are several shops who have already moved all their new development to 7.

The "use a previous version feature" significantly raises _my_ expectations and, I suspect, significantly raises the difficulty for SV. But there do not seem to be that many of us using this feature. Well, I'm one of them and many of my plans for C7 revolve around it.

Also, as I know many of the folks at SV, I have little question that they will get where I need them to be. And they will do it faster than previous managment.

Posted on Monday, August 17, 2009 by Frederick Doyle

That is indeed an interesting question, as I paid for the CSP, installed C7EE, and have not been able to use it EVEN AFTER learning the new features and layout. I agree with the MASK issue. I agree with the clunkiness, which is the most polite word I can come up with. I am VERY tempted to uninstall C7EE and move over to (god forbid) .NET, which has finally nicely implement a template-based approach. One thing is certain: if these problems are not taken care of QUICKLY, I will NOT pay for another CSP.

I understand the need for Softvelocity to make money and survive, but I also thinks that it's important to make a decent product, which they appear to (now) be failing at. After all, the push for the CSP subscription was for the "GOLD" version, not iron pyrite.

Posted on Monday, August 17, 2009 by Steven Parker

Frederick, my intention was not to slam SV or to denigrate C7. I am fully confident that I will soon be abandonning the C5.5 and C6 IDEs (iow, I'll learn the new IDE as thoroughly as I did the CDD IDE).

On way or the other, C7 _is_ the future of Clarion desktop development. Of that, I am convinced. The only question is "How soon?"

As to moving to .Net, well, that's another issue and there are lots of good environments out there to select among.

Posted on Tuesday, August 18, 2009 by Frederick Doyle

Thanks for your response. As a loyal SV customer from way back (CPD for DOS), I have been patient with them in the past, and truth be told, I have made a lot of money using their products. My only concern is paying to get this incomplete upgrade, when they said it was finished. Realistically, the money is not the problem, it was the expectation that certain functionality would, well, function. I'm still here though, am I not?

;)

Posted on Thursday, August 20, 2009 by Robert Wagner

Good article! Only one thing troubles me. Your list of 'show stoppers' is different from my list. And I suspect that other readers will have other lists. What's worse, we would all generally agree with each others list.

There should be a check box, when submitting a problem report to SV, that says 'Show Stopper', and we should be able to browse all such problems. That won't happen, but it sure would be nice.

So, Dave, that's where you could come in<g>. Put up a page for C7 (and another for Cnet), where we could all post our particular 'show stoppers'. Maybe they should be divided into development, and run-time, show stoppers.

My pet C7 show stopper right now is a run time bug concerning menus - a menu list which must overflow to additional columns loses 7 menu items (on my machine, at least). My end users are confused enough when they can see everything; it would be mayhem if I had to tell them that it's there, but you just can't see it<g>. It was reported months ago.

Posted on Thursday, August 20, 2009 by Robert Wagner

Good article! Only one thing troubles me. Your list of 'show stoppers' is different from my list. And I suspect that other readers will have other lists. What's worse, we would all generally agree with each others list.

There should be a check box, when submitting a problem report to SV, that says 'Show Stopper', and we should be able to browse all such problems. That won't happen, but it sure would be nice.

So, Dave, that's where you could come in<g>. Put up a page for C7 (and another for Cnet), where we could all post our particular 'show stoppers'. Maybe they should be divided into development, and run-time, show stoppers.

My pet C7 show stopper right now is a run time bug concerning menus - a menu list which must overflow to additional columns loses 7 menu items (on my machine, at least). My end users are confused enough when they can see everything; it would be mayhem if I had to tell them that it's there, but you just can't see it<g>. It was reported months ago.

Posted on Thursday, August 20, 2009 by Robert Wagner

Good article! Only one thing troubles me. Your list of 'show stoppers' is different from my list. And I suspect that other readers will have other lists. What's worse, we would all generally agree with each others list.

There should be a check box, when submitting a problem report to SV, that says 'Show Stopper', and we should be able to browse all such problems. That won't happen, but it sure would be nice.

So, Dave, that's where you could come in<g>. Put up a page for C7 (and another for Cnet), where we could all post our particular 'show stoppers'. Maybe they should be divided into development, and run-time, show stoppers.

My pet C7 show stopper right now is a run time bug concerning menus - a menu list which must overflow to additional columns loses 7 menu items (on my machine, at least). My end users are confused enough when they can see everything; it would be mayhem if I had to tell them that it's there, but you just can't see it<g>. It was reported months ago.

Posted on Thursday, August 20, 2009 by Steven Parker

Thank you.

>Your list of 'show stoppers' is different from my list.

I think that's to be expected.

Posted on Friday, August 21, 2009 by Dave Harms

Robert,

Interesting idea, thanks. I've added that as a feature request in FogBugz for the next version of ClarionMag, currently under development.

Dave

Add a comment

Clarion Magazine

Creating An XML RSS Web Site Summary With Clarion 7

by Steven Parker

Published 2009-08-08

With the collapse of the economy, I've found myself with rather a lot of time and not much to do. I started spending a lot of time on Netflix "Watch Instantly." As many of you know, I am very interested in the exegesis of obscure Biblical passages and in Biblical history and archaeology. As it turns out, there were several very engrossing documentaries on Biblical archaeology available to "watch instantly on your computer."

So, I did. Two, sometimes three films a day (admittedly, some of the films I "instanted" were commercial films and not especially "biblical;" there aren't that many Biblical archaeology documentaries made; lots of "faith and spirituality" films – which interest me not at all – but not that many of historical and none of exegetical interest).

To prevent becoming a complete workstation potato (that's the developer's variant on "couch potato" and involves a desk chair in place of a sofa), I decided to collect these film titles and summaries in one place. I could then pass the information on to like-minded ... folks.

I also decided to write up sermons I have delivered over the years – I had a lot of time on my hands – and, together with the film reviews, set up a web page. The web page would also contain links to other research or esoteric material to which I'd been referring people. This would allow me to keep all the materials in one place and minimize the opportunity to lose or misplace things. I could also make the information available to others without having to constantly lug around a notebook (the paper kind, not the kind with a keyboard).

Of course, one thing led to another and soon I had quite a collection of material. I had more material than I was willing to publish all at once (and, as a result, I haven't gotten around to writing up my sermons!).

The problem? How do I inventory the material? And, how do I keep track of what has been published (and when) and what has not?

Managing article inventory

The obvious answer was to make a list in a word processing document. Right. That worked for about a week.

The ideal solution is a small database. The main file would contain information like:

- Article ID
- Article title
- Article description (actually a blurb)
- Date posted to the web site

At this point, I don't actually need the blurb. It's one of those fields I tend to reserve for personal use. When I look at the record, the title may not be enough to bring the article's contents to mind. A description helps me remember. I also decided to make the database support multiple web sites (in one management app) on the off chance that I might ever want to index PAR2. So I also need

- Web site ID reference

And consequently I need a separate look up file for the web sites. This Sites file needs:

- Site ID
- Site Name
- Site Description

You can see that I've constructed a fairly standard parent-child structure here. Site ID links the Sites file to the Articles file, allowing by-site filtering. The Articles file has keys on ID (unique, primary, auto number) and date. The Sites file has keys on ID and Name.

This simple file structure allows me to create site-specific reports on:

- Items that have been posted and when they were posted
- Items that have not been posted (no posting date or a future posting date)

and

- Items that have not been posted yet (i.e., items with a forward reference in the date field)

(The second and third reports also filter on date.)

I particularly like the last-mentioned report. I can work with a report of unposted items to set my publication schedule. Then, if I enter these forward dates in the Articles record, the last report becomes my work schedule, telling me what to add to the finished web pages and when.

Ideally

Ideally, each item to be posted would be stored, locally, in a separate file (or possibly in a blob attached to the record). The type of file, HTML, text, doesn't really matter. When I want to update a site, I could run a Process, select the site ID and publication date, and create the final web page(s) from the source files for upload to the web server.

Of course, I'd have to distinguish the various kinds of sources I use: text, simple links, pictures.... But, that's another story.

I could even add a field to the Articles file for a Category, corresponding to a sub-page. This would allow me to organize the site into multiple pages.

As you may have inferred from Dave Harms' recent Clarion Magazine Roadmap, Clarion Magazine is probably built this way (though I am reliably informed that CMag's editing process starts with Word documents as its input before ending up with HTML). But I don't have the advantage Dave has of having mastered and written all the server-side web application code or even of having complete control over my server.

A fly in the ointment

The only caveat to this approach is HTML's reserved characters. Programming languages have reserved words. HTML has reserved characters. These are <, >, &, ‘ and ". If any of these characters are used in the text stored in my database, displaying that text will cause problems.

These characters need to be escaped. Some time ago, I wrote code to slice an incoming string from a data file and make the appropriate substitutions (< becomes &lt; and > becomes &gt; etc.) in writing the final string passed to HTML (or written to an HTML file). If I recall correctly, this was for a version of my knowledge base back around C4. Carl Barnes has also provided a function to escape these reserved characters in a note attached to Dave Harms' XML For Clarion Developers.

Unfortunately, I didn't have the time to make my new site this automated. This sort of architecture, entirely appropriate to a site as deep and complex as CMag, does not seem warranted by my current endeavor.

Setting up an RSS feed

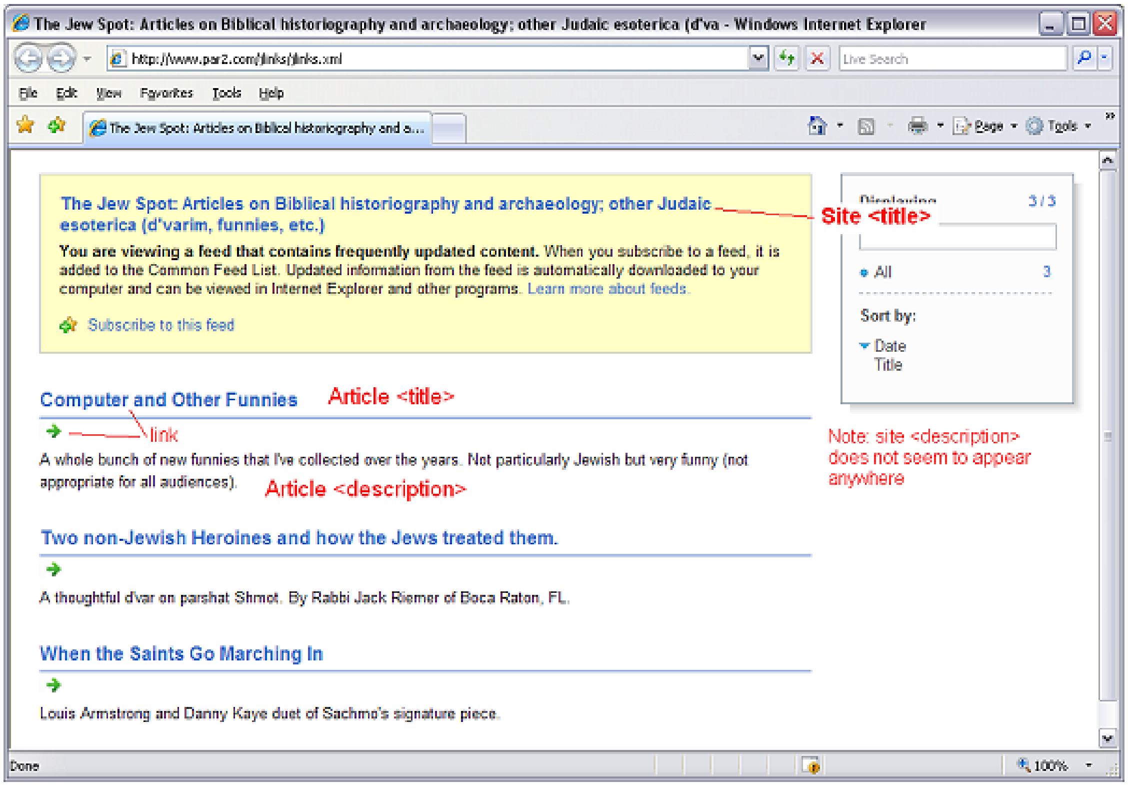
Dave Harms did a pair of articles on creating the XML file required for an RSS back in June 2003 (see Creating An XML RSS Web Site Summary With Clarion 6; I will assume you've read those articles and, therefore, will not go over the structure of the XML files which he described so well). In those articles, he pointed out that creating an RSS feed eliminated (or reduced) the need to email people ("subscribers") when a site changes.

I want this feature. And, even if most of my readers are unlikely to use it, it certainly makes the site seem more professional ("if you can't bowl ‘em over with brilliance, baffle 'em with ...," you get the idea).

Because Dave has done the leg work, I figured setting this up would be a no brainer. Well, that didn't turn out to be quite the case.

First, I needed some additional fields in both my files. Articles needed a URL field if I want people to be able to click on a link and read the article.

Examining the structure of RSS feed files, that description field I put in the Articles file for personal use turns out to be important. It is displayed in the page the user sees when they click on an XML link. See Figure 1.


The Jew Spot: Articles on Biblical historiography and archaeology; other Judaic esoterica (d'va - Windows Internet Explorer
http://www.par2.com/links/links.xml
Live Search
File
Edit
View
Favorites
Tools
Help
The Jew Spot: Articles on Biblical historiography and a...
Page
Tools
Displaying 3 / 3
All 3
Sort by:
Date
Title
Site <title>
Article <title>
link
Article <description>
Done
100%

The Jew Spot: Articles on Biblical historiography and archaeology; other Judaic esoterica (d'varim, funnies, etc.)

You are viewing a feed that contains frequently updated content. When you subscribe to a feed, it is added to the Common Feed List. Updated information from the feed is automatically downloaded to your computer and can be viewed in Internet Explorer and other programs. Learn more about feeds.

Subscribe to this feed

Computer and Other Funnies

A whole bunch of new funnies that I've collected over the years. Not particularly Jewish but very funny (not appropriate for all audiences).

Two non-Jewish Heroines and how the Jews treated them.

A thoughtful d'var on parshat Shmot. By Rabbi Jack Riemer of Boca Raton, FL.

When the Saints Go Marching In

Louis Armstrong and Danny Kaye duet of Sachmo's signature piece.

Note: site <description> does not seem to appear anywhere

Figure 1. Page displayed on clicking an XML link with sources

The Sites file needed:

- Copyright owner

- Site title
- URL

and

- Default XML file name

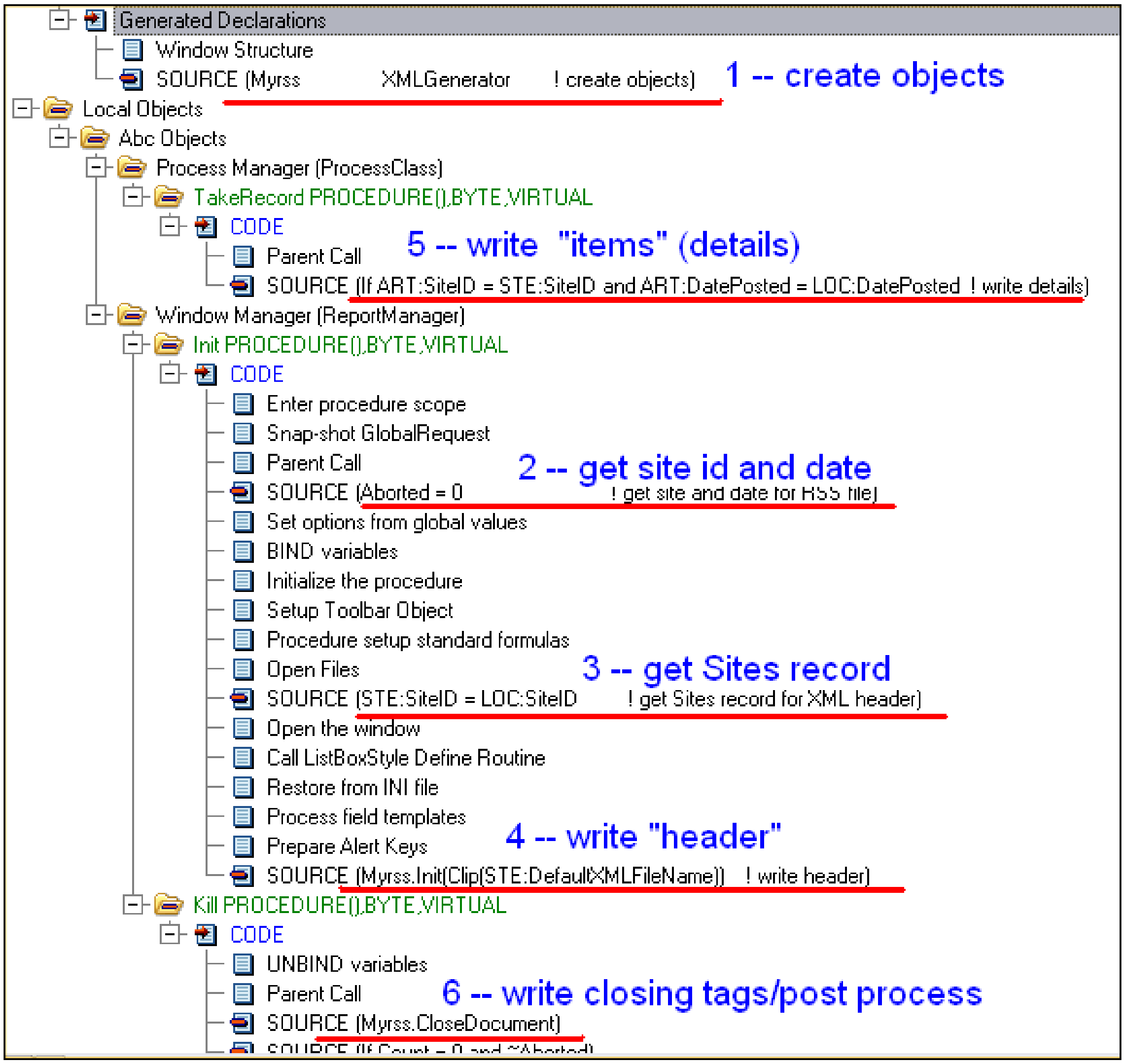
The article title and description fields are required for the feed file (again, see Figure 1, above). Use of the URL field is obvious. The Default file name field means that, when I create the file, it will always create with the same name (no typos). The feed will not work correctly if the XML file name varies from one site update to another.

Building the file

Start by creating the classes (After Global INCLUDES):

```
include('abprxml.inc'),once
include('anyascii.inc'),once
```

Then create a Process procedure on the Articles file. With this done, it takes only six embeds to create the final XML file.



```
  Generated Declarations
    Window Structure
    SOURCE (Myrss          XMLGenerator       ! create objects)      1 -- create objects
Local Objects
  Abc Objects
    Process Manager (ProcessClass)
      TakeRecord PROCEDURE(),BYTE,VIRTUAL
        CODE
          Parent Call                                                5 -- write  "items" (details)
          SOURCE (If ART:SiteID = STE:SiteID and ART:DatePosted = LOC:DatePosted  ! write details)
    Window Manager (ReportManager)
      Init PROCEDURE(),BYTE,VIRTUAL
        CODE
          Enter procedure scope
          Snap-shot GlobalRequest
          Parent Call                            2 -- get site id and date
          SOURCE (Aborted = 0                        ! get site and date for RSS file)
          Set options from global values
          BIND variables
          Initialize the procedure
          Setup Toolbar Object
          Procedure setup standard formulas
          Open Files                                 3 -- get Sites record
          SOURCE (STE:SiteID = LOC:SiteID          ! get Sites record for XML header)
          Open the window
          Call ListBoxStyle Define Routine
          Restore from INI file
          Process field templates
          Prepare Alert Keys                     4 -- write "header"
          SOURCE (Myrss.Init(Clip(STE:DefaultXMLFileName))    ! write header)
      Kill PROCEDURE(),BYTE,VIRTUAL
        CODE
          UNBIND variables
          Parent Call                        6 -- write closing tags/post process
          SOURCE (Myrss.CloseDocument)
          SOURCE (If Count = 0 and ~Aborted)
```

Figure 2. Embed usage

1) Create the required objects in Local data (instantiate the classes):

```
Myrss       XMLGenerator       ! create objects
Myascii     AnyAsciiFileClass
text        CSTRING(1000)           ! datum for final kludge
```

2) Early in Init, I get the Site ID and publication date for the filter:

```
Aborted = 0
Clear(STE:Record)
GetParameters(LOC:SiteID,LOC:DatePosted)  ! get filters
```

Aborted is a variable I use for two purposes: (i) did the app user (mostly likely, the site administrator) cancel the process, (ii) did the user get thrown out on an error. I use this at the end of the procedure.

And I make sure the user selected both a site ID and a date:

```
If ~LOC:SiteID or ~LOC:DatePosted
  Aborted = 1
  Return Level:Fatal
End
```

3) After Open Files, I fetch the Sites record. This record is necessary because some of the information that goes into the XML file comes from the Sites file.

```
STE:SiteID = LOC:SiteID ! get site for XML header
Access:Sites.Fetch(STE:SiteIDKey)
```

Standard stuff, so far. You may notice that there is no error check. This is because the GetParameters procedure ensures that there really is a matching record in the Sites file.

4) The heavy lifting begins at the end of Init, where I initialize the XMLGenerator class and write the header information (the channel description block). Note that nothing is hard coded, file variables are used throughout:

```
Myrss.Init(Clip(STE:DefaultXMLFileName))    ! write header
Myrss.OpenDocument('rssxxxxxxxxxxxxxxxx')
Myrss.XMLVersion = '1.0'
Myrss.AddXMLHeaderAttribute('encoding','UTF-8')
Myrss.AddTag('channel',",0)
Myrss.AddTag('title',Clip(STE:SiteTitle),0,'channel')
Myrss.AddTag('description',Clip(STE:SiteDescription),0,'channel')
Myrss.AddTag('language','en-us',0,'channel')
If Upper(STE:URL[1:4]) <> 'HTTP'
  STE:URL = 'http://' & STE:URL
END
Myrss.AddTag('link',Clip(STE:URL),0,'channel')
Myrss.AddTag('copyright','copyright ' & Clip(**STE:CopyrightOwner**) & |
   Year(TODAY()),0,'channel')
```

Regarding that second line: in his articles, Dave noted that SV's XMLGenerator class did not properly deal with attributes on root tags. In 2009, it still doesn't. So, the kludge Dave created then will still be necessary. (Robert Paresi's iqXML provides a createAttribute method that handles this and the subject is covered in the help. Personally, I blame Dave for "innocently" – so he says – leading me down a path away from iqXML. I think I'll have to rectify that mistake. (Dave's response: Um, Steve, that article was published six years ago, before iqXML, which in any case was first made available via Par2....))

Also note that I enforce URLs beginning with HTTP (at line nine). RSS validators will refuse to validate URLs that do not begin with HTTP. More, if you don't validate and just post the file, the link will not work. Best to be sure.

5) The Items are written in TakeRecord:

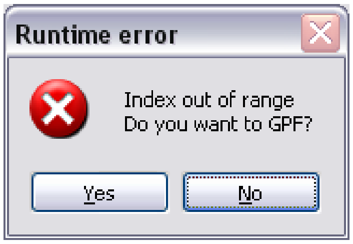
```
If ART:SiteID = STE:SiteID and ART:DatePosted = LOC:DatePosted  ! write details
  Myrss.AddTag('item','',0,'channel')
  Myrss.AddTag('title',Clip(ART:Title),1,'item')
  ! make this a CDATA section because the xml
  ! generator doesn't seem to escape reserved characters
  If Upper(ART:Url[1:4]) <> 'HTTP'
    ART:URL = 'http://' & ART:URL
  END
  Myrss.AddTag('link',Clip(ART:URL),0,'item')
  Myrss.AddTag('description',Clip(ART:Description),|
    1,'item')  ! also make this CDATA
  Count += 1
End

If Count > 15                           ! RSS limit
  MESSAGE('RSS may contain only 15 Items.'&|
     '||You are attempting to add more than that.'&|
     '||Processing is being truncated at 15 ....', |
      'Too Many Nodes', ICON:Exclamation)
   Return Level:Fatal
END
```

There are a number of interesting things happening here.

I start by wrapping Item section creation in my filter conditions. This was strictly for convenience. The proper place is in the template's default Filter prompt.

The next interesting bit is Myrss.AddTag('title',Clip(ART:Title),1,'item') (and the similar bit for the description tag).

What I found was that the XMLGenerator class was throwing errors on invalid array values when writing out the file. There was no helpful information in the message (see Figure 3, below). I finally tracked it down to the use of some of those reserved characters in my article titles and descriptions. (Actually, it was less "tracking down" and more WAG'ing – it seem that the XMLGenerator class may not escape reserved characters. Debugging this code is not a trivial exercise because the error is not posted in TakeRecord but only when the final XML file is written out, after the process is done.)



Runtime error

Index out of range
Do you want to GPF?

Figure 3. Helpful error received from XMLGenerator

Researching the AddTag method (abprxml.clw) showed that it supported CDATA. CDATA means Character DATA. This keyword tells the parser not to parse the data. (The same file in 9056 also shows support for CDATA.) My alternative was to preprocess the title and description strings, as discussed earlier. Adding a "1" parameter value to the method call seemed easier than:

```
DisplayString = FixUp(ART:Title)
Myrss.AddTag(‘title',Clip(DisplayString),0,'item')
```

where FixUp is my old escaping procedure (not to mention that I would have to find that old code). And, XML files created with the CDATA declaration pass validation (I used Feed Validator for Atom and RSS) and display correctly on my site. So, it's easy; it worked; I'm done.

Next, I count how many items have been added. Not only do I want to know that there were items added, RSS feeds have a 15 item limit. So the last thing I do in TakeRecord is to check that I am within that limit.

6) Finally, in Kill:

```
Myrss.CloseDocument ! write out the file
```

The XML file's contents are actually written to disk (this is where I was getting the array value errors) and the file closed by the CloseDocument method.

If the user did not cancel and there were no errors, but there were no records, I present a message:

```
If Count = 0 and ~Aborted
  Message('No qualifying record were ' &|
   'found.','Warning',ICON:Exclamation)
Else
  Myascii.OpenFile(STE:DefaultXMLFileName)
  Myascii.Read(text)
  Myascii.Read(text)           ! get second line of file
  If text = '<rssxxxxxxxxxxxxxxxx>'
    Myascii.Replace('<rss version="0.91">')
    Loop
      If Myascii.Read(text) <> level:benign
        Break
      End
      If text = '</rssxxxxxxxxxxxxxxxx>'
        Myascii.Replace('</rss><!--ignoreme-->')
      End
    End
  End
  Myascii.CloseFile()
End
```

If there were records, I use Dave's kludge to rewrite the root tag. Like Dave, I chose to use RSS version 0.91. This seems to be the most widely supported version of RSS.

Making it work

A small thing Dave left out of his article was "How to make it work." How do I create the link that calls the page shown in Figure 1 that invites user subscription?

Well, to make a long story short, it turns out that an XML file can be a link in an HTML href:

```
<a href="http://<domain>/<fileName>.xml">
  <img border="0" src="xml.png" title="Subscribe"
  width="36" height="14"></a>
```

Who'd'a thunk?

Also, the XML organization requests use of XML.PNG for the link image. What the heck, it's a free image ... I've included in the downloadable file at the end of this article.

Summary

C7 wasn't much help creating the web site. Well of course it wasn't. C7 doesn't do web pages.

However, the C7 app did make managing the site quite a bit easier. A fairly straight forward database application allows me to track my inventory, both published and yet-to-be-published, get reports and set my work schedule.

With the one caveat that the XMLGenerator classes don't allow adding attributes to root tags, I can just press a button and get the XML file I need for an RSS feed. I've eliminated variation in producing this file; once I've validated output from my little program, any XML file created will be a valid RSS feed file.

Next time: what it was like building the management app in C7.

Download the source

Steve Parker started his professional life as a Philosopher but now tries to imitate a Clarion developer. He has been attempting to subdue Clarion since version 2007 (DOS, that is). He reports that, so far, Clarion is winning. Steve has been writing about Clarion since 1993.

Reader Comments

Posted on Monday, August 10, 2009 by Mark Riffey

Steve,

If you use feedburner.com (free), it will automagically convert your feed into whatever format is necessary (Atom, RSS 1.2.345677, whatever) when an app connects to consume it.

Has nothing to do with Clarion per se, just thought Id slap a little spicy mustard on your XML sandwich.

Mark

Add a comment

Clarion Magazine

Writing Your Own Template Chain: Understanding Embeds

by Dave Harms

Published 2009-07-27

In Part 1 of this series I talked about the possibilities of generating non-Clarion code using the Application Generator and a custom template chain. As I noted in that article, templates have been written in the past for PHP, ASP.NET, and even for Java. But custom templates aren't really that common because the barriers to entry are fairly high. The shipping ABC and Clarion template chains are complex, and it isn't always easy to see what's going on.

In this series I've stripped down the templates to their essentials. Last time I talked about the #Template, #System, #Application, #Module and #Procedure statements, which represent the bare minimum "pieces" you need to assemble a custom template chain. I showed how code generation is handled when you use the embeditor, and I explained how to create simple embed points using #Embed. I also showed how to generate code into those embed points.

I'll get to some of the advanced uses of embed points shortly, but first I'd like to take a moment to compare embed points to the nearest analogue in the .NET world: partial classes.

Embeds and partical classes: a diversion

Embed points are, I think, absolutely crucial to effective application-scale code generation. I know there's been some discussion among Clarion developers using .NET about how the latest .NET platform and language improvements make embedded code unnecessary. Usually the argument revolves around something called *partial classes*.

Partial classes are a pretty simple concept. In Clarion (and Clarion#), you define a class like this:

```
AClass           CLASS,PUBLIC
AMethod              PROCEDURE
AnotherMethod          PROCEDURE
                 END



AClass.AMethod      PROCEDURE
  CODE
  ! Do something

AClass.AnotherMethod     PROCEDURE
  CODE
  ! Do something
```

But in Clarion# you have the option of adding the ,Partial attribute to the class. Partial lets you take your Binford 6200 chain saw and cut your class down the middle (or anywhere else you please) and put each part in its own source file. Here's a Clarion# version of the class, part 1:

```
APartialClass      CLASS,PUBLIC,PARTIAL
AMethod                PROCEDURE
                 END

APartialClass.AMethod      PROCEDURE
   CODE
   ! Do something
```

And here's part 2:

```
APartialClass      CLASS,PUBLIC,PARTIAL
AnotherMethod          PROCEDURE
                 END

APartialClass.AnotherMethod PROCEDURE
   CODE
   ! Do something
```

The compiler will take these two (or more) partial classes and combine them into a single class. Any code that calls this class will only see a single class. The only difference between partial classes and ordinary classes is the convenience of being able to split the class into multiple source files.

And that convenience is exactly where partial classes and embeds appear to cross paths. If you've done any work with Windows Forms in Clarion# (or another .NET language) you'll have seen partial classes in action. For any one form you have a single class, in two parts. One part (let's call this Part A) is under the control of the form designer, and you don't normally see the source code at all; instead, you see the same window the source code will create at runtime.

When, in .NET, you add a control to a Windows form, the designer adds the necessary code to the class, Part A. But when you tell the designer you want to respond to an event on the form, as when a user clicks a button, the designer generates a method declaration into Part B, and some code that calls that method into Part A. Part B is the bit that you can edit. You can add your own code to that method and it won't be overwritten by the designer.

So, this would seem to take care of embed points, right? I mean, partial classes are a brilliant little bit of engineering that make it easy to combine generated and non-generated code.

Partial classes are so brilliant, in fact, that they approach the functionality of, oh, Clarion for Windows version 1.0 Beta 3.

Where we came from

In 1994 the London Development Centre was hard at work on the first Windows version of Clarion. I'd somehow persuaded Ross Santos to include me as a co-author on his new Clarion book, and we were eager to get our hands on the new Application Generator.

That first version of the AppGen was an evolutionary improvement on CDD and CPD. It generated procedures, but in a monolithic way. You'd have one procedure template for each type of procedure. Ross and I got busy writing about the new AppGen.

And then the LDC released a new beta with a completely different approach to application generation, and Ross and I had to scrap most of what we'd written. Which is why I remember how this all happened.

Leaving aside the fact that CW 1.0 was strictly a procedural language, the first version of the CW AppGen was a lot like partial classes, with the embed list corresponding to one part of the class and the window formatter corresponding to the other. But even that early AppGen did something the Windows Forms designer doesn't: it embodied customizable logic. You could use template prompts to radically change the generated code.

It didn't take long for the LDC folks realized that it was going to be too much work to write one procedure template for each type of procedure developers might want to create, so they took a completely different approach. They broke down the procedure template into smaller pieces including, most importantly, #Control and #Extension templates. The idea was that you could drop a control template onto a window and have that control template embody all the necessary code generation logic. And you could drop multiple controls, or one control multiple times (if it was written to allow that) or any combination thereof.

There are a couple of interesting points here having to do with how control templates actually work. One is that there are three contexts for code generation. You use the #Generate statement to trigger generation, and you pass it one of three symbols: %Program, %Module or %Procedure. %Program is a single-valued symbol representing the #Program section of the template. Both %Module and %Procedure are multi-valued, and correspond to the currently selected module or procedure (usually obtained via a #For loop).

Let's take the case of a #Generate(%Procedure) statement. The #Procedure itself contains two kinds of information: stuff that gets generated, and stuff that doesn't. Stuff that doesn't get generated includes #Prepare and #AtStart sections as well as procedure prompts.

Now, what happens if you add a control template to a procedure? There is no #Generate(%ControlTemplate) statement. Instead, it's up to the control template to insert code appropriately using #At statements. And #At statements point to #Embed statements.

But the control template isn't just limited to the embeds in the procedure in which it appears; it can generate code into module and program embed points as well. For instance, the ABC BrowseBox control template includes this snippet:

```
#AT(%CustomGlobalDeclarations)
  ... a bunch of template code

#ENDAT
```

The %CustomGlobalDeclarations embed is located in Program.tpw, in the template's #Program section.

Similarly, the browse control embeds code in the procedure's %DataSection embed:

```
#AT(%DataSection),PRIORITY(3500),↵
  DESCRIPTION('Browse View & Queue for ' & %Primary)
#FIX(%Control,%ListControl)
#INSERT(%ConstructView)
#EMBED(%BeforeConstructingBrowseQueue, ''),%ActiveTemplateInstance,HIDE
#INSERT(%ConstructQueue)
#ENDAT
```

There's something particularly interesting in this block of code. Not only does the control template add a view definition to the procedure's data section (via the inserted %ConstructView group) but it also *inserts an embed point*. Think about that. The control template has just created a new location within its own generated code where other code can be added. This embed point has the Hide attribute, meaning it's only meant to be used by other templates, but there's no reason this couldn't be an embed point the developer can use.

Extension templates, by the way, have similar capabilities to control templates. They can place code into all sorts of embeds, and they can also generate new embeds. They just don't have any associated controls.

Embedding tricks

Not only can you take advantage of embed points at various places in the template chain, but you an also generate new embed points inside existing embed points which other template code can then use.

And the reverse is true, as Bruce Johnson recently reminded me. Templates can generate code into embed points only if those embed points exist. But if the embed points don't exist, you don't get an error. You might think you'd want an error in that situation (and there have been times I'd have saved myself a lot of trouble if the AppGen had pointed out I was trying to generate code into a non-existent embed), but in fact ignoring the error can be very useful. Because there's no dependency on the existence of embed points you can write a template that generates code that only comes into effect when yet another template is added to the procedure.

But as Bruce also said in that conversation, wait, there's more:

> Not only can one template inject code into another, but it can "be" the other, and run template code *as if* it was written in that template. It's OOP Jim, but not as we know it. (Ok, it's not OOP, but it's a form of Inheritance, one of OOP's three pillars.) In other words, using #CONTEXT, it's possible for one template to *extend* another template, as if the other template was running. All the variables, settings and so on visible to the other template become visible to you. Pretty much no-one uses #CONTEXT because it makes your head spin, but it's a fantastic tool for allowing templates to interact with each other.
>
> For example, a template can, in code, change the values of its prompts. So, even though you are asking the developer to set specific settings, the template code can keep an eye on the settings and adjust them if they're set inappropriately. Couple this with the #CONTEXT command, and it's possible for one template to set the user-entered fields for another template. So you might be adding some functionality to a browse, and you want to populate the ICONS tab of the browse settings with some pre-defined icons. No problem using #CONTEXT.
>
> Control and Extension templates don't just extend the base template chain. They're able to extend other Controls and Extensions. And you can write template code, as an extension, that can interact with the other extensions already on the window. It is this ability that makes it possible for so many commercial templates to exist with such a high degree of compatability. And indeed it is this ability which allows new functionality to be added to existing programs with such a high degree of confidence.

Summary

The Clarion template system is a toolset for building discrete pieces of an application and then combining those pieces in flexible ways while still allowing developers to add their own custom code. Although I've focused on control templates this time, much of what I've said applies to extension templates. Most of the real power of the template language is centered on embed points and control and extension templates.

This article has been a little heavier on the theory so I don't have an update to the template chain example from last time. In the next article I'll add some dynamic embed creation examples to that chain.

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors (ASJA).

Reader Comments

Posted on Friday, August 07, 2009 by JC Harris

I am titillated (if that's the right word) by your allusions to the possibilities of #CONTEXT. I would welcome more articles about -that-.

Add a comment

Clarion Magazine

Review: Clarion 7 Build 5768

by Dave Harms

Published 2009-07-15

Clarion 7 build 5768 follows the pattern of previous builds and includes some 80 bug fixes; I've reproduced the list at the end of this article.

There aren't any spectacular changes this time around, just a continued steady pace of bug fixes. And reports from the field vary widely, again as usual. Some developers consider C7 unusable, others find it nearly flawless and love it. Others use it but not yet in production. Your mileage, as the saying goes, may vary.

I've also heard of a few developers using C7 for development but creating C6 output. Certainly that's the entire purpose of the backward compatibility subsystem; you take advantage of the new IDE while still using your existing C6 (or earlier) compiler, runtime library and add-on products.

Owner drawn menus are affecting a lot of folks, it seems. C6 menus are standard Windows menus. If they aren't pretty enough for you there are third party products that will spiff them up. In C7 the menus are owner drawn, meaning the Clarion RTL decides on the menu's appearance. Aside from the fact that not everyone agrees with SoftVelocity's idea of what the menu should look like, the owner drawn approach means third party products can't change the appearance of the menus as they can in C6. Andy Wilton suggested that SoftVelocity provide a switch so developers can choose whether they want C6-style OS menus or owner drawn menus.

SoftVelocity has said something will be done to avoid displaying the window text editor by default; that change didn't make its way into build 5768, nor did the MASK fix.

The learning curve

There are lots of little things to be learned about the new IDE. Robert Paresi asked why it takes so long to make repeated column changes in the dictionary editor (as when changing the description field for a bunch of columns). If you change one description field, then select another field, you'll get a dialog box asking you: "The item has changed. Do you want to save changes?" and you have to click yes or press an appropriate key (or keys). It's a lot of mousing around, and I've noticed the same issue.

Jorge Lavera mentioned using the Save toolbar button to avoid getting the dialog box, so I tried that. Whoa. Now the column editing window closed, and I had to double-click to bring a column back up for editing. This was worse! But wait. I *wasn't* clicking on the Save Table button (the one with the floppy disc icon), I was clicking on the Save and Close button (the one with the green checkmark). Clicking on the Save Table button worked as Jorge described, and in fact I found it quicker to hit Alt-F,S with my left hand to save the changes before selecting the next column. So really it wasn't that painful a process after all, and the tabbed view of the editor definitely makes navigation easier.

SV's Diego Borojovich pointed out that all toolbar buttons are shortcuts to menu items, so look at the menu item to see the keyboard shortcut.

The Application Pad is going to take a bit of education as well. It's all too easy to see the Solution Explorer as the standard way of loading, generating and building applications, but in fact you should be dong this with the Application Pad. Only the Application Pad may not be readily apparent, depending on how you have your IDE configured. As I've said in just about every recent review, the AP is fantastic for template development: I can load, generate, and unload a test app in one click.

Release notes for Clarion 7.0.5768

Fixes/Changes/Features

PTSS 20447: Line X/Y written incorrectly
PTSS 20448: Behavior change
PTSS 20553: Radio control in window formatter
PTSS 20704: Buttons not showing as MultiLine in Designer
PTSS 30532: a preliminary fix for 800000xx colors in Designer
PTSS 30073: Structure locks up IDE
PTSS 31504: Control Template Pad - Populating without dragging
PTSS 31794: Make file list selection for Individual File Overrides show in alphabetical order
PTSS 31841: A heap corruption error was sometimes thrown when trying to convert a file
PTSS 31848: Wallpaper converted incorrectly
PTSS 32116: Move a button from tab to window did not show warning message.....
PTSS 32130: Offset throws error message when it should default to 0
PTSS 32448/33598: Template Editor code folding didn't create region for upper case/lower case functions
PTSS 32499: CE characters in listbox header
PTSS 32634: TRN on STRING not working in report designer
PTSS 32734: CE characters in listbox header
PTSS 32737: STRING gets serialized wrong if "VariableString" property is set to FALSE
PTSS 32761: No EntryPattern (MASK) property available on controls.
PTSS 32906: Control template pad scroll to bottom
PTSS 32990: Windows lose controls, loss of standard clarion behaviour
PTSS 33051: Center Justification not working properly
PTSS 33099: Report formatted numeric display
PTSS 33203: Cancel to add a break Group doesn't work
PTSS 33211: Wrong reproduce of some Cyrillic symbols in Window Designer
PTSS 33344: Listboxer Formatter loses Euro-Sign (
PTSS 33351: Wrong icons on Spin box
PTSS 33385: checkbox and transparent in designer
PTSS 33488: Window painter ignores DCT field settings
PTSS 33502: Populate fields on tab
PTSS 33530: Justification not working properly
PTSS 33545: Cannot create browse control on a tab
PTSS 33553: Unexpected "duplicate symbol" error could be reported at link time
PTSS 33653: Manual Entry of ICON does not work
PTSS 33661: Report formatted numeric display

PTSS 33677: Quick ClassBrowser dropdowns empty
PTSS 33686: Changing the properties of Window changes size of entry controls that are on a TAB
PTSS 33698: Saving window properties TILED Wallpaper
PTSS 33699: Problem with adding the options to an option group
PTSS 33716: No Keystroke Exit on Edit Picture
PTSS 33718: Template Editor code folding missing some block statements
PTSS 33729: Tab Indent with dialog, one line too many
PTSS 33738: Block Indent does not save indent value
PTSS 33754: Can not open window designer with sub menu
PTSS 33774 version resource template problem
PTSS 33786 version resource template problem
PTSS 33805: When you add a string to a window and assign it a use variable the screen's picture should change to display the use variable.
PTSS 33827: Cannot drop a column from Datapad on to a Box control
PTSS 33828: Populating a line on a report has a default value of 1000 for both Width and Height
PTSS 33858: an exception was thrown if the redirection file had a line like *.lib = ..\otherdir
PTSS: 30062: Pre and Post build events had did not work for Win32 projects
PTSS: 3702: finding fields in files can fail

CHANGE: Template Editor code folding improved speed of parsing.
CHANGE: C%V%HTM%X% is now added to the initial list of lib files to ignore (LibIgnoreList.xml) when attempting to copy dlls.
CHANGE: Win32 project files would return relative paths rather than absolute paths causing problems with reloading solutions

FIX: An exception was thrown if you moved the location of an installed version of Clarion and then tried to use that version without updating the options for that version
FIX: Any controls can be added in Option in code. Validator does not catch it.
FIX: Designer: Arrows for "Spin" control were shown not correctly on the form.
FIX: Background for Toolbox equals Background for Window.
FIX: Controls not visible in similar location as on a SHEET
FIX: DATE_TIME was incorrectly listed as a valid data type when editing application variables
FIX: Error occurs if Tabs add in the Sheet by means of Property "Tabs" - Collection.
FIX: If property Disable is True for Sheet then it will not be visible on the form.
FIX: Image of BackgroundImage for Window is not correctly placed on the form.
FIX: Memory associated with the redirection system was not released as soon as possible when you edited a redirection file or loaded a file using File/Open File using Redirection.
FIX: Menubar and Toolbar can be moved with the the cursor in TOA.
FIX: Minimize of Sections works incorrectly.
FIX: Mode "Cnter" for BackgroundImage of control Toolbar works incorrectly.
FIX: Populating LIST on a SHEET is incorrect parent
FIX: Property "Scrollbars" for Combo and List works incorrectly.
FIX: Some controls change size if some properties for the Window change.
FIX: Switching Band Views - missing functionality
FIX: The debugger did not load/use the redirection system properly when debugging products built with other versions of Clarion other than the current version
FIX: The debugger would crash if you attempted to debug a program that could not be loaded into memory

FIX: The position of the RadioButton does not change after copying from one Option into another.

FIX: The vertical line at the left in a Page Layout View for Report should not be moved if scrolling.

FIX: Unhandled exception has occurred, if value the AT attribute is not integer.

FIX: Arrow buttons are active for Page Header, Detail, Page Footer and Report Form in TabOrderAssistant

FIX: Forbid input of negative values for some properties List Box Formatter for "Combo"

FIX: incorrect message in the status bar when the app was unloaded.

FIX: The SQL drivers could cause a Function Sequence Error if you issued a SELECT statement using PROP:SQL on a table that had a primary key that had a key component that was a TIME, DATE or timestamp field

---

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors (ASJA).

Reader Comments

Posted on Thursday, July 16, 2009 by Kraig Whiting

The problem I still have with C7.5768 is that my C6 apps still don't port over automatically. It takes me 1/2 hour to go through all the procedures making the minor changes required. By the time I finish the conversion, it's time to get back to the "real" work. So I'm just not spending enough time in C7 to really understand and appreciate.

By the time I get back to C7 again, I've made more changes to my C6 programs and the process starts over.

Yikes!

Add a comment

Clarion Magazine

Clarion and .NET: The 50,000 Foot View

by Dave Harms

Published 2009-07-10

For some time now I've been working on a rewrite of the Clarion Magazine web site. I'm doing this in .NET, and it's my first really big .NET development effort. Along the way I'm trying to make use of leading edge tools, practices and methodologies as much as possible, and that's led me to begin organizing my thinking about the various ways there are to develop applications, whether for Win32 or .NET. In particular, I begun documenting several high level views of the development process.

High level views are sometimes described as "50,000 foot" views, as when you look down on something from a great height and can (hopefully) see how all the pieces fit together. In this article I'll present two 50,000 foot views of Windows development, one for traditional Clarion Win32 programming and one for .NET programming.

These outlines aren't complete; I add to them regularly. And you can expect to see some of this kind of outlining applied to a future version of Clarion Magazine. Stay tuned....

I'm presenting these views partly to illustrate the major differences between Win32 development and .NET development, and partly to address the question of whether a port of an existing Clarion application to .NET is viable.

The Clarion (Win32) 50,000 foot view

For purposes of this article I've divided my 50k views into two main areas: the development environment and the application architecture. The development environment is the tool or set of tools developers use to create the application; the application architecture is the set of organizing principles behind the code developers create. (Other possible areas of consideration include the application design process, the database design process, and SQL.)

The luxury Clarion developers have enjoyed over the years is having both the toolset and a ready-made application architecture in one package. That makes it easy to get started developing applications. But eventually you need to start understanding more about what's going on under the hood, and that becomes doubly important as you consider .NET development.

So here's my 50k view of traditional (Win32) Clarion development:

1. Development environment
   - a. Development Tools
     - i. Clarion IDE
       1. Data Dictionary
       2. AppGen
     - ii. Clarion add-on products
   - b. Key language/platform details

- i. Data types
- ii. The Windows API
- iii. The Clarion RTL
- iv. Procedural code
- v. Classes and interfaces
- vi. Threading
- vii. Error handling

- c. Code generation
  - i. Template language
- d. Best practices
  - i. Naming conventions
  - ii. UAC
  - iii. Debugging
  - iv. Use of ABC
  - v. Profiling
  - vi. Build automation
  - vii. Avoiding code smell
- e. Reporting
  - i. Built-in reports
  - ii. Report writer
  - iii. Third party products
- f. Distribution
  - i. Installation
  - ii. Code signing
- g. Languages
  - i. Clarion
  - ii. C
  - iii. C++
  - iv. Assembler
  - v. Modula-2

2. Application architecture
   - a. Data access
     - i. FILE grammar
     - ii. VIEW structure
     - iii. ABC classes
   - b. Business logic
     - i. Validation
     - ii. Custom (domain) logic
   - c. User interface
     - i. Window designer
     - ii. Report designer

iii. Browse/form

If you've worked with Clarion for at least a few years (as most of us have) then there won't be much on that list that's new to you. If had to guess I'd say the least-known items would include dealing with UAC, build automation (which isn't quite as easy to do in Clarion as it is in .NET) and code signing.

The application architecture section of the outline covers the key elements of data access, validation, and the user interface (menus, browses, forms and reports). I'll come back to the architecture a little later on, as there are some significant differences between the traditional Clarion approach and current thinking about business application architecture.

The .NET 50,000 foot view

The .NET 50k view is, predictably, just a little bit more complex. There are several reasons for that. One is that software development has evolved considerably since Bruce Barrington laid down the foundations for Clarion. Another is that the .NET framework provides a much richer API than the Windows API. And finally, while traditional Clarion programming largely targets the desktop, .NET programming covers desktop, mobile and web platforms.

1. Development environment
    a. Development Tools
        i. Clarion.NET
        ii. Clarion 7 (custom templates)
        iii. Visual Studio
        iv. VS add-ons
    b. Key language/platform details
        i. Data types
        ii. Managed code
        iii. The .NET framework class library
        iv. Classes and interfaces
        v. Exceptions
        vi. Threading
        vii. Reflection
        viii. Events and delegates
        ix. Generic types
        x. Lambda expressions
        xi. Serialization
        xii. Interop
        xiii. LINQ
    c. Code generation
        i. Code level
        ii. Application level
        iii. Tools
            1. Clarion
            2. MyGeneration
            3. CodeSmith

4. Other

d. Best practices

i. Design patterns
ii. Separation of concerns
iii. Design by contract
iv. Domain-driven design
v. Fluent interfaces
vi. Test-driven development
vii. TDD and Mocking
viii. Debugging
ix. Profiling
x. Naming conventions
xi. Documentation
xii. Build automation
xiii. Avoiding code smell

e. Reporting

f. Distribution

i. Obfuscation
ii. Installation
iii. Code signing

g. Languages

i. Clarion#
ii. VB.NET
iii. C#
iv. F#
v. IronPython
vi. IronRuby
vii. Boo
viii. C++/CLI
ix. Javascript (web UI)

h. Language comparisons/differences

i. Clarion vs Clarion#
ii. Clarion/Clarion# vs C#
iii. Clarion/Clarion# vs VB.NET
iv. C# vs VB.NET

i. The .NET framework

i. Windows Presentation Foundation (WPF)
ii. Windows Forms
iii. ASP.NET
iv. Communications and Workflow
v. Fundamentals

- - - vi. Data, XML and LINQ
  - j. Funky new stuff
    - i. Oslo
    - ii. Sketchflow
  - k. Metrics
    - i. Code coverage
    - ii. Products
      1. NDepend
      2. NCover
  - l. Operating systems
    - i. Virtual machines
    - ii. Windows
    - iii. Linux (mono)
  - m. Design methodologies
  - n. Design tools
    - i. UML

2. Application architecture
   - a. Separation of concerns
     - i. Tiers and layers
     - ii. Testing
     - iii. Reuse
   - b. Data layer
     - i. Direct access (ADO.NET)
     - ii. Object-relational mapping (ORM)
       1. NHibernate
       2. Entity Framework
       3. Commercial offerings
     - iii. Query tools (LINQ)
   - c. Business logic
     - i. Validation
     - ii. Custom (domain) logic
   - d. User interface
     - i. WinForms
     - ii. ASP.NET WebForms
     - iii. ASP.NET MVC
     - iv. WPF
     - v. Silverlight
       1. (see also WPF)
   - e. Application frameworks
     - i. WinForms
       1. Clarion# (unknown - ABC#?)

2. CSLA
3. StrataFrame (proprietary)
4. Mere Mortals (mm.net)
5. DevExpress eXpressApp Framework (proprietary)
6. NConstruct

- ii. WPF/Silverlight
    1. Caliburn
- iii. Web
    1. ASP.NET MVC
    2. S#arp (web)
    3. ASP.NET
    4. Spring.NET
    5. Castle Monorail
    6. BFC (commercial)
    7. DotNetNuke
    8. Morfik (commercial)

f. Web services

g. Configuration/extension

- i. Inversion of Control (IoC) frameworks
- ii. Managed Extensibility Framework (MEF)

h. Platforms

- i. Windows
- ii. Other via Mono (Linux, Mac, ...)

i. Design patterns

- i. MVC
- ii. MVP
- iii. MVVM

The first six items under the development environment section have a lot of parallels with the Clarion list. There are more IDE options than I've listed, but Clarion.NET and Visual Studio are the ones you're most likely to encounter.

There are some hugely important differences under the key language/platform details heading. Here's where you see the evolution of programming languages writ large. In .NET, exception handling is the norm rather than error handling. Reflection is a massively important addition, underlying many .NET features. Events and delegates likewise are something every .NET developer should understand. Generic types improve code flexibility and re-use; lambda expressions make code more configurable; and the list goes on.

Code generation in many ways is more primitive in the .NET world than it is in the Clarion world. I've yet to see an end-to-end code generation system for .NET that manages the entire application creation process from conception to implementation. Instead, code generation often focuses on the smaller pieces of the puzzle, and is more often a one-off process rather than a system that interacts with the developer and allows custom code to be easily interwoven with generated code.

While there are some informal best practices for Clarion (don't use implicit variables, if you're duplicating code write a template, keep your routines small, etc.) I don't see anywhere near the kind of agreement and formal description of best practices as exists in the .NET world. Hot topics include separation of concerns (more on that later), test-driven development (of which I'm a big fan), design by contract, etc.

As for reporting, I'm told by a recognized Clarion expert in the field, who's also done a lot of .NET work, that Clarion's reporting engine (which is in Clarion#) is still a superior tool for business reports.

Distribution of .NET applications may require the additional step of obfuscation, since IL code is a whole lot easier to reverse engineer than object code.

I find the comparison between multi-language support quite interesting. Back when Bruce Barrington went shopping for a new Clarion compiler he came across the good folks at JPI, who had the idea of creating multiple languages that all compiled to the same back end code. It was a brilliant idea, even if JPI was never that successful. So brilliant that .NET does exactly the same thing. All .NET languages compile to the same underlying IL code, just as JPI languages compiled to the same object code.

But JPI was a wee little thing and Microsoft is the 800 lb gorilla, so the fact that you could mix Modula-2 or assembler with Clarion never mattered as much as it did that JPI's binaries weren't directly compatible with Microsoft's binaries, thanks to a different parameter passing convention.

.NET seriously levels the playing field; I use a lot of different libraries, most of which I suspect are written in C#, but as long as they work, the language they're written in is immaterial to me. But really the languages of choice for most developers will come down to Clarion#, C# and VB.NET.

In the past VB.NET has gained some features ahead of C#, and vice versa. Microsoft has more recently stated that it doesn't favor C# over VB.NET (and regrets giving the impression that C# was the "standard" .NET language); in the future both languages will get the same features more or less at the same time. So far Clarion# has support for generics, lambda expressions, etc. As you'd expect there's some lag between Microsoft releasing a feature in C#/VB.NET and SV adding that feature to Clarion#; happily SV doesn't have to reverse engineer new features; since all .NET languages generate readable IL code it's straightforward to verify the Clarion compiler's output as compared to, say, the C# compiler's output.

But .NET isn't just a bunch of compilers; it's also a huge library of classes. It's mind-bogglingly big. Lots of stuff you used to have to write code to do, or buy a library to do, is right there in the framework.

Some of the more important areas of the framework, from the perspective of business software development, include:

- WinForms - A user interface library for "traditional" windows applications. In other words, it's a nice wrapper around the WinAPI/GDI code.
- ASP.NET - Web application development support
- Windows Presentation Foundation - A DirectX-based user interface featuring richer controls and a cleaner separation between UI and business logic
- Silverlight - a subset of WPF for browsers (and a Flash competitor)
- XML libraries
- ADO.NET - Database access
- Language Integrated Natural Query (LINQ)

Now, how do all these pieces come together in a .NET application?

.NET application architecture

Clearly the toolset for .NET development is far richer than it is for Win32 development. But having all of those power tools is one thing; using them to build a useful, reliable and maintainable application is another. And that's where the subject of application architecture comes in.

A traditional Clarion application is pretty much a monolith, a single entity with everything wrapped up in one. Yes, you can create a multi-DLL application, and you can put your ABC library in one DLL etc. but there really isn't a formal separation between the different parts of the application. It's not uncommon to see a single browse procedure containing business logic and direct file access. But why is that a bad thing?

Layers and separation of concerns

One of the most powerful forces driving the evolution of software development is an idea called the "separation of concerns" or SoC. You may think that's just another buzzword of the day, but it's been around for at least 35 years, which is forever in computing terms.

The idea behind SoC is that there are important benefits to splitting up an application into distinct features. From a business software development point of view, this usually means creating at least the following distinct areas within a program:

- A data layer that communicates with the data store (i.e. an SQL database)
- A business logic layer that embodies the behavior unique to the application
- A presentation layer that provides a user interface

Often the business layer is further subdivided into a business layer and a service layer, but that topic is well beyond the scope of this article.

Clarion ABC applications tend to partially separate the data layer via the FileManager and RelationManager classes, although as noted it's easy to interact directly with the database and bypass these classes. And the business logic is quite often encoded within the procedure itself; when you write business logic in a procedure embed, there's a good chance you're really placing it inside an instance of the WindowManager. Once it's embedded inside that WindowManager it's much more difficult if not impossible to reuse elsewhere. Yes, templates can help with the duplication problem, but they're not a complete answer because they don't address testability.

One way to look at the drive for SoC is to think of it as a way to separate the business logic from the other parts of the application. If you can separate that logic so it isn't *dependent* on a particular user interface and/or data store then you can much more easily test and reuse that code. The user interface may look pretty, and the back end database may be fast and capable, but the business logic is where your application lives and dies.

If there's one big idea you should take away from this article it's that teasing apart your application into its smallest functional units pays huge dividends in testability and reuse. You can do that now in Clarion. But it's one of those things that's generally easier to achieve in .NET.

SoC - the reality

The reality is that while SoC is a terrific idea with a huge payoff, it isn't something you see in every .NET application, at least not yet.

It's quite possible to write a WinForms application using a very Clarion-like architecture; in fact it can be difficult to keep the business logic out of the UI layer. ASP.NET WebForms has a similar problem; although you *can* achieve a high degree of SoC with WebForms, the standard approach leads many developers to embed a lot of code, which is no more reusable or testable than Clarion embed code.

WPF, which is becoming more and more popular, has built-in support for SoC, and ASP.NET MVC does the same for web development, so there are alternatives that foster SoC.

.NET application frameworks

At the beginning of this article I said that Clarion developers have for many years enjoyed the luxury of a single, ready-made application architecture (in two flavors, Legacy and ABC). .NET developers, at least at present, don't have one universally accepted choice, a situation complicated by the possibility of needing to develop a code base to support desktop, mobile and web applications.

There are quite a number of application frameworks that purport to make the job of writing .NET applications easier. I've listed some of these; none to my knowledge have the all-inclusive nature of a Clarion template chain.

Speaking of which, it's been some time since SoftVelocity gave guidance on the features Clarion developers can expect in a .NET template chain. That information (several years old) was on the old blog site which is no longer available.

Porting to .NET

There is a significant gulf between the architecture of a typical ABC or Legacy Clarion application and a "best practices" .NET application with distinct layers. That doesn't necessarily mean a port is impossible; SoftVelocity has ported the Clarion runtime library, and there's no reason why they couldn't port, say, ABC as well. The major difference between Clarion and Clarion# is in the window handling; rather than a WINDOW structure with an ACCEPT loop Clarion# uses the standard .NET approach of creating an instance of System.Windows.Forms.Form and processing user input with event handlers (which really isn't that different from how the WindowManager does things).

If it becomes possible to port an existing ABC app to Clarion#, then I think it's reasonably safe to say that will not be a multi-tier application with separated concerns. And that might be an acceptable compromise for many developers. For anyone willing to refactor their code base for greater flexibility and reliability, a multi-tier template chain will clearly be a better option.

Summary

While there are some parallels between .NET and traditional Clarion Win32 programming, there are also some important differences, as the two 50k views I've presented here demonstrate.

Architecturally, the biggest shift since the original Clarion templates/models were conceived (other than the growth of object-oriented languages) is the move toward separating the business, presentation and data layers. As well, developers now have the option of targeting web, mobile and desktop development while mixing and matching languages.

Although SoftVelocity has not given a clear indication on whether it will support porting existing applications to .NET, it seems to me that porting existing applications to Clarion# is feasible, although it's unlikely such a port would result in a multi-tier application exhibiting a clear separation of concerns.

By way of review, here's a side-by-side look at the two views. If you think of items that should be added to either of these lists, please post a comment below.

Clarion Win32	.NET

1. Development environment
    - a. Development Tools
        - i. Clarion IDE
            - 1. Data Dictionary
            - 2. AppGen
        - ii. Clarion add-on products
    - b. Key language/platform details
        - i. Data types
        - ii. The Windows API
        - iii. The Clarion RTL
        - iv. Procedural code
        - v. Classes and interfaces
        - vi. Threading
        - vii. Error handling
    - c. Code generation
        - i. Template language
    - d. Best practices
        - i. Naming conventions
        - ii. UAC
        - iii. Debugging
        - iv. Use of ABC
        - v. Profiling
        - vi. Build automation
        - vii. Avoiding code smell
    - e. Reporting
        - i. Built-in reports
        - ii. Report writer
        - iii. Third party products
    - f. Distribution
        - i. Installation
        - ii. Code signing
    - g. Languages
        - i. Clarion
        - ii. C
        - iii. C++
        - iv. Assembler
        - v. Modula-2

1. Development environment
    - a. Development Tools
        - i. Clarion.NET
        - ii. Clarion 7 (custom templates)
        - iii. Visual Studio
        - iv. VS add-ons
    - b. Key language/platform details
        - i. Data types
        - ii. Managed code
        - iii. The .NET framework class library
        - iv. Classes and interfaces
        - v. Exceptions
        - vi. Threading
        - vii. Reflection
        - viii. Events and delegates
        - ix. Generic types
        - x. Lambda expressions
        - xi. Serialization
        - xii. Interop
        - xiii. LINQ
    - c. Code generation
        - i. Code level
        - ii. Application level
        - iii. Tools
            - 1. Clarion
            - 2. MyGeneration
            - 3. CodeSmith
            - 4. Other
    - d. Best practices
        - i. Design patterns
        - ii. Separation of concerns
        - iii. Design by contract
        - iv. Domain-driven design
        - v. Fluent interfaces
        - vi. Test-driven development
        - vii. TDD and Mocking
        - viii. Debugging
        - ix. Profiling
        - x. Naming conventions
        - xi. Documentation
        - xii. Build automation
        - xiii. Avoiding code smell
    - e. Reporting
    - f. Distribution
        - i. Obfuscation
        - ii. Installation
        - iii. Code signing
    - g. Languages
        - i. Clarion#
        - ii. VB.NET
        - iii. C#
        - iv. F#
        - v. IronPython
        - vi. IronRuby
        - vii. Boo

	viii. C++/CLI ix. Javascript (web UI) h. Language comparisons/differences i. Clarion vs Clarion# ii. Clarion/Clarion# vs C# iii. Clarion/Clarion# vs VB.NET iv. C# vs VB.NET i. The .NET framework i. Windows Presentation Foundation (WPF) ii. Windows Forms iii. ASP.NET iv. Communications and Workflow v. Fundamentals vi. Data, XML and LINQ j. Funky new stuff i. Oslo ii. Sketchflow k. Metrics i. Code coverage ii. Products 1. NDepend 2. NCover l. Operating systems i. Virtual machines ii. Windows iii. Linux (mono) m. Design methodologies n. Design tools i. UML
2. Application architecture a. Data access i. FILE grammar ii. VIEW structure iii. ABC classes b. Business logic i. Validation ii. Custom (domain) logic c. User interface i. Window designer ii. Report designer iii. Browse/form	2. Application architecture a. Separation of concerns i. Tiers and layers ii. Testing iii. Reuse b. Data layer i. Direct access (ADO.NET) ii. Object-relational mapping (ORM) 1. NHibernate 2. Entity Framework 3. Commercial offerings iii. Query tools (LINQ) c. Business logic i. Validation ii. Custom (domain) logic d. User interface i. WinForms ii. ASP.NET WebForms iii. ASP.NET MVC iv. WPF v. Silverlight 1. (see also WPF) e. Application frameworks i. WinForms

        1. Clarion# (unknown - ABC#?)
        2. CSLA
        3. StrataFrame (proprietary)
        4. Mere Mortals (mm.net)
        5. DevExpress eXpressApp Framework (proprietary)
        6. NConstruct
    ii. WPF/Silverlight
        1. Caliburn
    iii. Web
        1. ASP.NET MVC
        2. S#arp (web)
        3. ASP.NET
        4. Spring.NET
        5. Castle Monorail
        6. BFC (commercial)
        7. DotNetNuke
        8. Morfik (commercial)
f. Web services
g. Configuration/extension
    i. Inversion of Control (IoC) frameworks
    ii. Managed Extensibility Framework (MEF)
h. Platforms
    i. Windows
    ii. Other via Mono (Linux, Mac, ...)
i. Design patterns
    i. MVC
    ii. MVP
    iii. MVVM

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors (ASJA).

Reader Comments

Posted on Sunday, July 12, 2009 by Rhys Daniell

We certainly seem to have lost our way since the visionary days of BB and crew.

I need a tool which enables me to deliver systems which support and enhance my customers' business, quickly and robustly.

I find it incredible that after 20 or so years of PC tool development by the best minds in the business, be they at Microsoft, Sun, SV, or in the public domain, it is actually harder and slower to produce a Windows business app than it was back then.

Much to the chagrin of my trendier developer friends I've been able to easily outperform them in the Windows arena by using C6 over SQL.

Supporting .Net is only playing catchup. If BB and team were at work today they'd probably give us something along the lines of a Javascript appgen for for a thin client with a NetTalk server supplying the data. Incredibly unfashionable but devastatingly effective.

Posted on Monday, July 13, 2009 by Bruce Johnson

While it's natural to compare the differences of Clarion to other development environments, we should also note that the goal here is not to make Clarion exactly like everything else. By definition that would render Clarion irrelevant.

Rather than search for ways that Clarion# can expose these additional levels of complexity, it's always been the Clarion way to rather hide this complexity.

In the time of CPD the IDE, as a code editor, was primitive by the standards of the day (Turbo Pascal). Likewise, CW 1.0 spent a large amount of effort in hiding the complexity of the Windows API (unlike some of the popular Windows tools which were then available.)

SV is a small shop by the standards of MS et al. But by most Clarion programmer standards they're huge. Indeed most Clarion programmers are either working alone or in very small teams. Likewise, the goals of most Clarion programmers are different to the goals of a 200 person programming team working in a large corporate.

As you pointed out one advantage Clarion has has is by having a very small number (two I would argue) of basic frameworks. This makes the code highly consistent across multiple programmers, and across many years of programming. While having a dozen frameworks to choose from may be an advantage to some I would argue it is also a millstone for others. One can see the fragmentation which is occurring in JavaScript, and the extent to which JavaScript scripts are either framework dependent, or incompatible to see the disadvantage of the fragmented approach.

If your goal is to pad your CV, working on technologies that will make you more desirable at your next job offer, then Clarion isn't (and I would argue, never has been) the right choice.

However if your goal is to write code, that gets out the door and makes money, code which is stable and which will remain maintainable for 5, 10 or 20 years, then Clarion has a proven track record. The bleeding-edge is seldom where the money is, and the market for good, solid, business apps isn't going away anytime soon.

Equally, there is a limit to how long technologies can live for. Clearly, with the smallest number of exceptions, we are not programming in DOS anymore. .NET may ultimately become as necessary to our end-users as Win32, HTML and so on are today. However I would venture that that tipping point has not yet arrived.

There's a clear split in the Clarion community which puts those with commercial off-the-shelf software somewhat at odds with those that build smaller, custom apps and consult on a different job every few months.

The former group has a huge investment in existing Win32 code, and all technology arguments pale compared to the desire to easily and simply port that code into .NET. They are selling to a market that neither knows, nor cares, what language the product is written in.

However the latter group needs to prove technical prowess both to gain new customers and to impress older customers who care about such things.

For example, in most typical TPS-to-SQL ports the primary goal is to get a working port out the door as soon as possible. Then optimizations can be applied where necessary to make better use of the "framework". In the same way an "immediate" port of Win32 to .Net, followed by a refactoring of code where desired is the only way large apps will ever make it into .NET. The option to spend years, or even months, on a rewrite before square one is reached is economically unfeasible, and possibly politically impossible.

For those needing to prove technical prowess to hook the next customer, there is a need to know all the latest terms, to be able to play nice with others, and to incorporate the latest buzzwords.

By the same logic the new, cutting edge stuff, will always get the most press. The old stuff is, well, old, and boring. We're all doing it so it's no longer newsworthy. This results in a skewed outlook, it seems like the new stuff is the most important as it oscillates from headline to headline. Unfortunately though the machine requires ever more newer and exciting frameworks and technologies to report on.

Speaking personally, I don't want a Clarion that "keeps up". I want a rock-solid Clarion into which I can write code that will still be running 10, or 20, years from now. And it is my fervent desire that Clarion# will deliver a long-term .NET experience rather than one which touches a dozen or more buzzwords but is forgotten as soon as the Next Big Thing comes along.

cheers
Bruce

Posted on Monday, July 13, 2009 by Rhys Daniell

Thanks Bruce, plenty of insightful comments from someone who probably knows more about the Clarion developer base than any other active community member.

Having recently tangled with my first .Net application - I let an experienced .Net programmer do the the heavy lifting - I'm aghast at what an awful environment it is, and how unsuited existing web tools are to the development of rich business applications.

.Net reminds me of Sapphire, RBase, and other long-forgotten systems that were around when I first discovered CPD - they demo-ed well, could put together 90% of a good application very quickly, and then hit various brick walls once you tried to produce something to meet specific requirements.

We need a rich thin client with a web server behind it. ClarioNet came very close, but as an orphan which now has a competitive sibling disputing parenthood it's not going to get the resources it needs any time soon.

At this stage the best solution is probably a Windows thin client, i.e. Terminal Server or a similar RDC-based solution. However, as with ClarioNet, the server resources required to scale to any significant size are eye-watering, with as few as 50 concurrent users practicable per server.

Part of the .Net experience was to find that a rich browser application was too much for the ancient hardware at the first serious customer site. (Murphy's law, alive and well.) So I rolled up my sleeves and created a Windows 'thin' client drawing data from a NetTalk server via XML and xFiles. I was able to replicate the functionality of the .Net application in a tenth of the time, deliver a rich experience for the end user, and require a fraction of the system resources on the client PC.

The main drawback is the requirement for a Windows client. I'd like to be able to provide the same functionality in a browser app. HTML simply doesn't allow the positional control required for a rich user experience, so that leaves Java (leave me out, I'm a Clarion programmer), or Javascript. Now, JS has been the poor cousin until recently but suddenly browsers are producing serious improvements in performance so maybe its time has come?

Posted on Monday, July 13, 2009 by Dave Harms

Rhys,

Sounds to me like you're talking specifically about the web aspect of .NET, not the framework as a whole. Web development has lagged desktop development for years, for obvious reasons. I'm a little biased, but I think the .NET web platform of choice these days, at least for new development, is Microsoft's ASP. NET MVC with JQuery (a Javascript library) for the rich UI.

Certainly there are some significant battles shaping up for rich internet apps, with Javascript, Flash and Silverlight all major contenders.

Dave

Posted on Tuesday, July 14, 2009 by Dave Harms

Bruce & Rhys,

Bruce, you've done a nice job of balancing the many forces at play at the intersection of Clarion and .NET.

If I may, I think the most important paragraph in the article is this one:

"If it becomes possible to port an existing ABC app to Clarion#, then I think it's reasonably safe to say that will not be a multi-tier application with separated concerns. And that might be an acceptable compromise for many developers. For anyone willing to refactor their code base for greater flexibility and reliability, a multi-tier template chain will clearly be a better option."

My intent in writing this article was to point out the rather large gulf between a typical Clarion Win32 app and a best-practices .NET application. Is this a gulf Clarion developers, at least those who will develop/are developing for .NET, wish to bridge? I think it's impossible to decide that without at least a minimal understanding of the issues.

Rhys, with all due respect a lot of things have gotten easier/better in .NET. But having better tools doesn't necessarily mean better code; it just means the opportunity is greater. And of course the greatest difference at the moment is that there is no .NET AppGen available yet, nor any accompanying templates.

The architectural decisions SV makes are worth watching, because they will shape the Clarion community for years to come.

Rhys mentioned that SV is only playing catch up by supporting .NET. Really, I can't think of a time when Clarion wasn't in that mode. Clarion 1.0 had a dongle, just as the industry moved away from that kind of copy protection. CPD needed LEMs to support file formats like BTrieve and to call C libraries, and it generated pseudocode, not compiled code, and needed an interpreter. CDD introduced a file driver system and a standards-based DOS UI, at a time when the industry was moving to Windows. CW 1.0 was a procedural Windows language, when the world was going object-oriented. And on and on.

I think it's inevitable that a small vendor is always going to be playing catch up. Small companies can't compete with big companies on all points; they have to choose their niche. And I'll argue that Clarion's primary niche (since CPD) has always been , and will always be, code generation.

The real question remains: what will that code generation look like for .NET, given the architectural possibilities?

Dave

Posted on Tuesday, July 14, 2009 by Rhys Daniell

Dave,

Yes, I'm talking about the .Net/web, I can't see any compelling reason to do WinForms in anything but Clarion (but perhaps I'm expecting too much from C7?).

I haven't seen any tools in the .Net world that match Clarion's functionality but there seem to be 3rd party addons for Visual Studio which could get you very close. Could SV be just too late to the party this time?

You're right, your observation about multi-tier development is a key statement. In our shop we find ourselves moving increasingly large chunks of business logic from the front end to the back end (in our case, MSSQL). There don't seem to be any really good tools to manage SQL code, maybe there's an opportunity there for SV?

Cheers
Rhys

Posted on Saturday, July 18, 2009 by Lee Vasic

What made Clarion great was that they didn't think and design stuff like everyone else. Now it seems like they want to be a 3rd party add-on supplier for Microsoft. It seems like Claron has passed the baton to Alpha Software.

Posted on Saturday, July 18, 2009 by Lee Vasic

What made Clarion great was that they didn't think and design stuff like everyone else. Now it seems like they want to be a 3rd party add-on supplier for Microsoft. It seems like Claron has passed the baton to Alpha Software.

Posted on Monday, July 20, 2009 by Alex McCullie

I'm a bit late reading your article, Dave, and well done for introducing some aspects of the .NET world to Clarion readers.

I've used Visual Studio 2003/2005 for sometime to develop intranet web apps for business. Just to clarify terms you are developing ASP.NET web applications with any .NET supported programming language. I use VB.NET.

Compared to the self-contained world of Clarion Win32, .NET feels wild and woolly. Inevitably a .NET developer will work at writing more lower-level code and creating architecturally more complex programs. You only have to think of the mix of web technologies to appreciate the difficulty.

To be fair to the complexity of .NET it is important to compare the underlying assumptions and objectives of the respective development enviroments. Clarion was originally designed to do one thing - produce self-contained, single-layer fileserver database applications. Clarion did that very well. In subsequent years as scaleability and other market forces became issues, Clarion was adapted to bolt on database server access still utilising the same technology. After teething problems this addition settled and became useful and reliable. Clarion, though, still produces all-in-one Windows application that assumes a constant connection with a database source.

.NET was initially built as an all-in-one replacement for Windows VB6 and C++ and web ASP developers. Since that time web concerns have dominated .NET development. For example, ASP.NET (via whichever .NET programming language) treats all data sources as disparate, disconnected and remote. This is typical of the web model of data access that Microsoft addressed with .NET and that they see as the future for business software development. Combine that with the desire for scaleability, reusability and interoperability of software components, especially in large-scale Enterprise work, you have a very different approach to software development.

Without doubt Clarion Win32 is still one of the quickest ways of developing self-contained Windows database apps and considerably faster than doing the equivalent in .NET. But that's not the primary focus of .NET.

If you want to get a feel now for a ASP.NET (web) business application in VB.NET or C#, download a copy of IronSpeed generator (http://www.ironspeed.com) on trial. In practice you'll need MS SQL Server and have installed .NET SDK (free). You won't need Visual Studio. You can generate an attractive full-feature web app in a couple of hours.

Again, Dave, thanks for the article.

Alex

Posted on Tuesday, August 04, 2009 by Dave Harms

Alex,

Thanks for the comments. I don't know that I'd agree that web concerns have dominated .NET development, but certainly WebForms is often seen as a better option for simple apps.

There are still lots of desktop apps that depend on databases, but increasingly that access is mediated by an object-relational mapping layer like Entity Framework or NHibernate. That's not disconnected in the sense of a web app, but it does abstract the database access even more than the ABC class library does.

Dave

Add a comment

Clarion Magazine

A Brief History of Clarion (Updated)

The following is a work in progress. If you have material you'd like to contribute to this history of Clarion, please send it along or post a comment at the bottom of the page (you must be logged in first).

Events with incomplete or uncertain information are colored yellow.

Date	Description	Notes
1973	Bruce Barrington starts HBO & Company.	Barrington writes a multi-user operating system and macro language for the IVPhase desktop computer, one key feature of which is memory-mapped video. BB forms ideas that would later evolve into Clarion.
1979	Niels Jensen starts Midas ASP	Develops and markets sofware for microcomputers. First product is WordIndex, an add-on for WordStar. Midas acquires an Irish shell company named Borland International Ltd. Eventually Midas begins doing business as Borland, and later Phillippe Kahn is hired to run the US operations.
1981	HBO goes public, BB lives "pretty comfortably"	BB buys an Apple II but finds Pascal frustrating. Doesn't consider this a business machine
1981	IBM releases the PC	BB says Aha! Memory mapped video! and calls up some HBO pals.
1983	Work begins on Clarion 1.0	Monty Shaw joins the team in late summer and persuades BB that development should be done in C. Bob Steinke is hired out of HBO but doesn't offically start until 1984.
1983	Niels Jensen and Poly Pascal	Jensen discovers a product for the CPM-80 called Poly Pascal, written by Anders Heilsberg. Borland acquires marketing rights and renames Poly Pascal to Turbo Pascal. Jensen's team later writes the enormously popular Sidekick.

1984	Work continues on Clarion 1.0	Bob Steinke is brought in to write the compiler and collaborate with BB on language design. Gary Liming starts a few weeks after Bob Steinke; Shaw, Liming and Steinke have known each other since grade school and become the "three musketeers" for much of 1984. John Herron is hired late in the year and helps recruit the rest of the core team, including Stephen and Kim Kemp and Tony Jackson.
1985	Borland moves to London	Begins developing new compiler line for Modula-2, C and C ++
1986	Clarion 1.0 released, April.	Clarion 1.0 is released at Spring Comdex in Atlanta. To combat copy protection, the first release of Clarion requires a hardware dongle (the "Actuator"), as do all apps created with Clarion. Unfortunately the industry is in the midst of a backlash against copy protection; negative press almost kills Clarion. BB reluctantly agrees to remove the Actuator in release 1.1.
1986	JPI is formed	Niels Jensen is also one of the founders of Borland (1983).
1987	Borland USA acquires Wizard C, which became Turbo C	The Borland language team leaves and forms JPI.
1987	Designer under development	Bruce Barrington and John Herron begin writing the Designer product. Steve Parker recalls BB saying John Herron was the creator of Designer: ("It took him a year and it almost killed him").
1988	CPD 2.0 released, May	The first version of Clarion to include Designer. Wins PC Magazine's Editor's Choice. InfoWorld rates CPD along with Paradox as the best programmable database.
1990	TopSpeed technology licensed	Niels and Bruce work out a deal to use TopSpeed compiler technology in Clarion 3.0 (CDD 3.0)
1990	Report Writer introduced	Approx date: "a year and a half after CPD"
1991	Euro Sales Manager	George Ratcliff of Essex is hired as the European Sales Manager
1991	CLARIONET	No, not the thin client technology for Clarion for Windows, but an online information service announced by Clarion Software as part of the Clarion Developer's Exchange, for just $150 per year. Ten hours of toll free access included!

1991	**DevCon #1**	The very first Clarion DevCon was held Sept 22-25 at the Westin Cyprus Creek Hotel in Ft. Lauderdale. CDD 3.0 announced, but little shown of substance. Keynote speakers included Bruce Barrington, Will Fastie, and Niels Jensen.
1991	Clarion Conference in Johannesburg, South Africa	November 12&13 at the VW Midrand Conference Centre. BB was present at this conference.
1992	Staff turnover	Six of the original seven R&D people, as of 1990, have left.
1992	**DevCon #2**	Marriott Harbour Beach in Ft. Lauderdale. CDD 3.0 shown, but not in final release yet. Jeff Duntemann the editor of PC Techniques was keynote speaker, this was presumably following his article in Dr. Dobbs Journal in 10/91 entitled "Sympathy on the Loss of One of Your Legs"
1992	CDD release at end of year (or might have been early '93)	
1992	JPI and Clarion Software merge in April, announcement made at the end of the year.	The new company has a head office in Pompano Beach and a development center in London (LDC). Later the LDC is revealed to be under contract to TopSpeed.
1993	Aussie DevCon	Sydney, Australia, CW 1.5 released, also "pen stuff" for CDD
1993	**DevCon #3**, Sept, first CW demo	Ft. Lauderdale (Marriott Harbour Beach again?). David Bayliss speaks and shows some actual Clarion for Windows applications. Ross Santos finds the third party vendor displays "boring". Keynotes by Will Fastie and Mark Burgess.
1993	DevCon Boston	Hosted by Ben Brady
1993	Tom Moseley hired to work on templates	Tom previously worked with Dave Howington on the LPM add-on.
1993	Clarion RADical Conference in Johannesburg	May 6&7 at the Fourways Indaba Hotel
1994	CW 1.0 Beta 2 released	Then called CDDW
1994	CW 1.0 Beta 3 released	Mid-year release, and the first to include the AppGen.

1994	Clarion Consulting Services formed	headed up by Chris Behling, former VP of Marketing
1994	Randy Wood hired as VP of Product Development	March 21, 1994
1994	**DevCon #4**	The Boca Raton Resort & Club. Keynote by John Dvorak (?) Outside events (sailing etc.) are cancelled due to the weather, attendees are not reimbursed for the extra money spent to keep spouses happy.
1994	DevCon West	May 7&8, Los Angeles
1994	Aussie DevCon	Sydney, Australia
1994	Clarion Software becomes Topspeed Corporation	
1995	DevCon West	John Griffiths recalls the DoubleTree hotel in '95 (confirmed by Stamos) and Bruce Barrington showing an early CW beta, presumably of 2.0.
1995?	South American DevCon?	Argentina?
1995/09	CW 1.5 released	
1995	CDD 3.1 released	
1996/01	Final CW 1.5 release	Version 1.502. Includes support for 32 bit executables.
1996	Kerrville gathering	Kerrville, TX. Pics. Ross Santos discovers fire ants.
1996/06	CW 2.0 Released	Adds object orientation, ActiveX support, enhanced SQL support
1996/09	**DevCon #5**	Held at the Hyatt Regency at the Orlando airport, and generally regarded as the absolute worst DevCon location.
1996	Arizona DevCon, Nov	Mesa Pavillion Hotel. Hosted by Kathryn & Phil Carroll, and I think Larry Teames had just a bit to do with it as well.

1997/01	Aussie DevCon, Feb	Sydney, Australia. The 1997 Oz DevCon was held at the Gazebo hotel, Sydney Jan/Feb and included Bruce Barrington, Tom Mosely, Scott Ferret etc. Internet Connect was shown and was very exciting. Who would have thought you could flick a switch and turn your desktop app into an internet app?? It was all going to be just soooo easy...
1997/03	Final CW 2.0 release	
1997/04	Euro DevCon	Held at the International Hotel in London. "A bunch of folks went to Beefeaters dressed up in costume. James Fortune was Friar Tuck....and a great time was had by all." - Skip Williams
1997/09	**DevCon #6**	Radison Bahia Mar Hotel, Ft. Lauderdale. Keynote by Peter Coffee.
1997/12	Clarion 4 released	Introduced the ABC class library and template chain. The "for Windows" was dropped from the product name. There was no version 3; the official reason was that "1.5 should have been 2.0" and so 2.0 should have been 3.0, but it's generally believed that the company wishes to avoid any numerical association with the CDD 3 debacle.
1998/02	Sydney DevCon	Sydney, Australia. The 1998 Oz DevCon was originally also to be held at the Gazebo hotel but this was changed and it was actually held the Millennium Hotel (not The President's Hotel) in late March. Presenters included Bruce Johnson (a much appreciated regular in Australia...), Erik Pepping, Scott Ferret and Gus Greces. A sales guy from Topspeed called Antonio Rajan managed to antagonise nearly everyone with his session but fortunately none of his "future directions" such as dumping the desktop development environment and becoming an "Internet company", and changing the name from Clarion ever came to fruition... well so far anyway...
1998	eData is formed	Hank Asher goes on to hire most of the London team.
1998/05	Clarion 5.0 released	First appearance of Professional and Enterprise versions

1998/05	Euro DevCon	Ramada Gatwick Airport
1998/09	**DevCon #7**	Marina Marriott hotel in Ft. Lauderdale. Keynote: Mr. Edward E. Iacobucci - Citrix chairman, chief technical officer and vice president of strategy and technology.
1998/05	ETC I	Lee White holds the first East Tennessee Clarion Conference in Gatlinburg.
1998/11	ConVic	Ballarat, Victoria, Australia. The first of Chris Livingstone's ConVic conferences
1998/11	South African DevCon	Johannesburg, Indaba Hotel, FourwaysCorrection. Richard Taylor was present.
1999/02	ClarionMag begins publication	
1999/03	Aussie DevCon	Sydney, Australia. Article.
1999/04	Euro DevCon	Amsterdam, Golden Tulip Barbizon Palace
1999/05	DevCon Latin America	Article
1999	Roy Rafalco resigns, Frank Watts hired as new CEO	
1999	ConVic '99	Mt. Buffalo, Australia. Article. Article.
1999/06	Clarion 5.0B released	Final 5.0 release
1999	Clarion 5.5 beta	
1999/09	**DevCon #8**	Hyatt Regency's Pier Sixty Six in Fort Lauderdale. Hank Asher keynotes, Main Street USA strategy announced, and it appears to many attendees that the company is being sold out in an attempt to grab a slice of the dot com pie.
200/05	SoftVelocity formed	TopSpeed's VP of Product Development, Robert Zaunere, heads a group of investors who purchase the Clarion product line from TopSpeed Corporation. The new company is called SoftVelocity Inc.
2000/05	ETC II	Lee White holds the second East Tennessee Clarion Conference in Gatlinburg.
2000	ConVic	Daylesford, Australia. Article.

2000/08	Clarion 5.5 released	Adds interfaces to the language
2000	ConDev Gescla, Brazil	Bonito - Mato Grosso do Sul
2001/10	South American ConDev	Argentina (last Argentinian Clarion conference before the economic collapse)
2001/03	ConVic 2001	Mt. Buffalo, Australia. Article.
2001	ConDev Gescla, Brazil	São Paulo
2002/01	Clarion 5.507 released	Final 5.5 release
2002	ConDev Gescla, Brazil	Florianópolis - Santa Catarina
2002	ConVic 2002	Geelong, Australia. Article.
2002/05	ETC III	Lee White holds the third East Tennessee Clarion Conference in Gatlinburg. Articles.
2002	ConDev Gescla, Brazil	Curitiba - Paraná
2003	ConVic 2003	Geelong, Australia. Article.
2003/11	Clarion 6.0 Gold	C6 Gold released. Major changes include support for full OS threads.
2004/05	ETC IV	Lee White holds the fourth East Tennessee Clarion Conference in Gatlinburg. Articles.
2004	ConDev Gescla, Brazil	Porto Alegre - Rio Grande do Sul
2004/05	Clarion 6.1	C6.1 Changes include XP theme support.
2004/09	**DevCon #9**, Orlando	Held at the Disney contemporary resort. First SoftVelocity DevCon. Some Clarion# vaporware shown. Articles.
2005/05	Clarion 6.2	New features include enhanced SQL functionality, new core templates, EIP, RTL and RTF improvements.
2005	ConDev Gescla, Brazil	Foz do Iquaco, Brazil. Article #1.Article #2.
2006/01	Clarion 6.3	C6.3 released.
2006	ConDev Gescla, Brazil	Bonito - Mato Grosso do Sul
2006/05	Aussie DevCon	Geelong, Australia. Articles.

2007/02	First Clarion 7 hand coder's release	Build 1457 was released on Feb 23, 2007
2007/05	Aussie DevCon	Sydney, Australia. Articles.
2007	ConDev Gescla, Brazil	São Paulo - São Paulo
2007/11	First Clarion# hand coder's release	Nov 17
2008/05	Aussie DevCon	Eden, Australia. Articles. First public appearance of AppGen.
2008/10	C7 AppGen Third Party Release	Oct 21
2008/12	C7 AppGen CSP release	Dec 24
2008	ConDev Gescla, Brazil	Juiz de Fora - Minas Gerais
2009/04	C7 Gold Release	April 13
2009/06	CONDEV GESCLA (Brazil)	June 11-14 in Curitaba

My thanks to Bob Steinke, Luis De Almeida, Geoff Robinson, Paul MacFarlane, Mike Hanson, Stamos Fafalios, David Jung, David Bratovich, Jose de Mello Junior and other unnamed Clarion folk for their contributions and corrections.

Larry Sand notes that the Internet Archive Wayback Machine shows the history of the TopSpeed web site until early 2001. Then the SoftVelocity web site is phased in late 2000.

Reader Comments

Posted on Monday, March 02, 2009 by Luis De Almeida

November 12&13, 1991: Clarion Conference in Johannesburg, South Africa, at the VW Midrand Conference Centre. BB was present at this conference.

May 6&7 1993: Clarion RADical Conference in Johannesburg at the Fourways Indaba Hotel. Richard Taylor (in charge of Documentation & Training, I think) was the Topspeed representative.

Posted on Monday, March 02, 2009 by Luis De Almeida

Correction: Richard Taylor was at the Johannesburg Clarion "Wizatron" Conference in 1998, not the Clarion RADical Conference in 1993! Maybe someone else can shed a bit more light on this...

Posted on Monday, March 02, 2009 by Larry Sand

The Internet Archive Wayback Machine shows the history of the TopSpeed web site until early 2001.
http://web.archive.org/web/*/http://www.topspeed.com

Then the SoftVelocity web site is phased in late 2000:
http://web.archive.org/web/*/http://www.softvelocity.com

Posted on Monday, March 02, 2009 by Geoff Robinson

The 1997 Oz DevCon was held at the Gazebo hotel, Sydney Jan/Feb and included Bruce Barrington, Tom Mosely, Scott Ferret etc. Internet Connect was shown and was very exciting. Who would have thought you could flick a switch and turn your desktop app into an internet app?? It was all going to be just soooo easy...

The next year the 1998 Oz DevCon was originally also to be held at the Gazebo hotel but this was changed and it was actually held the Millennium Hotel (not The President's Hotel) in late March. Presenters included Bruce Johnson (a much appreciated regular in Australia...), Erik Pepping, Scott Ferret and Gus Greces. A sales guy from Topspeed called Antonio Rajan managed to antagonise nearly everyone with his session but fortunately none of his "future directions" such as dumping the desktop development environment and becoming an "Internet company", and changing the name from Clarion ever came to fruition... well so far anyway...

Posted on Tuesday, March 03, 2009 by Paul MacFarlane

I believe the '95 devcon was in Los Angeles at a Hotel near LAX. I was trying to get there but had a conflict.

Posted on Thursday, March 05, 2009 by Daniel Scott

DevCon4 The Boca Raton Resort & Club

Clarion pisses off all the wives when outside events (Sailing Etc.) are cancelled due to the weather. We were never reimbersed for the extra money we spent to keep our wives happy. An omen of things to come when paying Clarion in advance?

Posted on Thursday, March 05, 2009 by Mike Hanson

One of the DevCon Wests was definitely at the Doubletree Hotel, which I think was near LAX. I'm not sure if it was the one in El Segundo, though.

Posted on Thursday, March 05, 2009 by Stamos Fafalios

You have me wrong, they were Florida Devcons I was thinking about, and yes it was the Westin (not Hyatt, I was wrong there)
I have the flyer for Devcon 92 in my hand, Jeff Duntemann the editor of PC Techniques was keynote speaker, this was presumably following his article in Dr. Dobbs Journal (which I have) in 10/91 entitled "Sympathy on the Loss of One of Your Legs" and I presume the photos in there are from Devcon 1.
I have the handout of the 1991 Conference with roster of attendees and 3rd party folks also flyer for Devcon 96.

There is a bunch of stuff here like a big article by Richard Aarons (Jan 87) from PC Magazine describling the Clarion language at the time of 1.1 and saying that in '87 Clarion will bring out a compiler able to produce executables <G>.

Give me a couple of weeks and I'll find you lots of stuff.

Posted on Friday, March 13, 2009 by David Jung

1994 DevCon #4? The Boca Raton Resort & Club. Keynote by John Dvorak.

I don't know if it was #4 or not, but I was at this DevCon with my wife. I happened to throw out the binder and conference materials about a month ago after having kept them up to this time. If I had thought about it, I would have asked if anyone wanted them...

Posted on Friday, April 03, 2009 by David Bratovich

DevCon West 1994 took place on May 7th & 8th.

Posted on Monday, April 06, 2009 by Dave Harms

Thanks David.

Dave

Posted on Tuesday, April 07, 2009 by Jose de Mello Junior

Hi,

In Brazil there is a group of users called GESCLA. We are organized since October/2000 performed as a first event for professionals who work with CLARION. Today we are organizing our 10th event, which will be held in the city of Curitiba, State of Paraná in the period from 11 to 14 June 2009, under the name CONDEV GESCLA, an event organized entirely by users.

City Year
2000 Bonito - Mato Grosso do Sul
Caraguatatuba 2001 - São Paulo
2002 Florianópolis - Santa Catarina
2003 Curitiba - Paraná
2004 Porto Alegre - Rio Grande do Sul
2005 Foz do Iguaçu - Parana
2006 Bonito - Mato Grosso do Sul
2007 São Paulo - São Paulo
2008 Juiz de Fora - Minas Gerais
2009 Curitiba - Paraná

Posted on Tuesday, April 07, 2009 by Dave Harms

Thanks Jose - I'll add that to the document.

Dave

Posted on Wednesday, April 08, 2009 by Stamos Fafalios

As promised, here are my Archives if they are of use:

DEVCONS

1st Devcon 91 at Westin Cypress Creek Ft Laud Docs Yes
2nd Devcon 92 at Marriot Harbour Beach Resort Docs Yes
3rd Devcon 93 at Docs Yes
4th devcon 94 at Boca Raton Resort & Club Docs Yes
Devcon 96 at Hyatt Regency Orlando Airport Docs Yes
Devcon 97 at Radisson Bahia Mar Ft Laud Docs Yes
Devcon 98 at Marina Marriott Ft Laud Docs Yes
Devcon99 at Hyatt Regency Pier 66 Ft Laud Docs Yes
Devcon 2004 at Docs Yes

Euro Devcon 97 at International Hotel london Docs Yes
Euro Devcon 98 at Ramada Gatwick London Docs Yes
Euro Devcon 99 at Golden Tulip Barbizon Amst. Docs Yes

ETC 98 Docs Yes
ETC 2000 Docs Yes
ETC 2004 Docs Yes

Devcon West 95 at Doubletree hotel LA Docs Yes

Arizona 96 CW Expo at mesa pavilion hotel AZ Docs Yes

PUBLICATIONS

The Clarion Technical Journal complete July/Aug 89 Vol1 No1 to July/Aug 95 Vol 7 No4, all good condition
also on CD

The Clarion for Windows Journal Vol1 No1 to Vol 1 No6, good condition

BoxSoft Banter 1 - 3

PPS !COMMENTS various 93 - 95

Clarion Herald 1.1 July 92 - 1.6 Jun 93

Clarionette V1 No1 to V1 No3

Clarion Developer V2 No9 to 1994, most issues

Topspeed Developer 1995 feb

Written Clarification (UK) 96 - 99 various

Posted on Thursday, April 09, 2009 by Dave Harms

Thanks Stamos! I'll update the document.

Dave

Posted on Friday, May 01, 2009 by David Jung

And to make this really interesting...where are the people named in the list now? :)

Posted on Friday, May 01, 2009 by Ray Rippey

I think somewhere around Clarion for Dos 3 Topspeed released a personal edition that was very inexpensive (I think around $99)... the idea was great to get folks introduced to Clarion and have it affordable so schools would buy it (if my memory serves correctly). It was a great little program that I think was wizard driven so you could make simple databases very quickly.

Anyone have more particulars on this?

Posted on Wednesday, May 13, 2009 by Tom Drum

Why no mention of when Bruce Barrington was about to abandon Clarion altogether before Bob Zaunere was kind enough to buy the rights to it in 2000?

Posted on Wednesday, May 13, 2009 by Dave Harms

Thanks Tom - Andrew Guidroz also mentioned this. I've updated the doc (see May 2000).

Dave

Posted on Tuesday, May 26, 2009 by David Bratovich

DevCon West 1995 took place on June 10th & 11th.
Location: DoubleTree LAX 5400 West Century Blvd.

Presented/Coordinated/Organized By DataForce
Bruce Gilham (President DataForce Los Angeles)

Co-Sponsored by...

TopSpeed

Mitten Software

The Clarion Tech Journal

Clarion User Groups of California

(LA/Modesto/Riverside/Sacramento/San Deigo/San Francisco/Silicon Valley)

Speakers/Trainers/Vendors...

Bruce Barrington

Dave Harms

Ross Santos

Andy Stapleton

Bob Foreman

Todd Seidel (ToolCraft)

Bruce Wojak (C3 Development)

Mike Hanson (BoxSoft)

Posted on Thursday, May 28, 2009 by Dave Harms

Thanks David - wow, that takes me back. I think that was the first time I spoke at a Clarion conference.

Dave

Posted on Thursday, May 28, 2009 by David Bratovich

Hey Dave,

Do you remember what you spoke about?

Off the top of my head, I can't remember the topic.

I do remember going to a Dodger game on the night of the Dinner Cruise. I also remember a discussion I had with Bruce B. about CD3 and OS/2.

Dave B.

Posted on Thursday, May 28, 2009 by Dave Harms

I'm pretty sure I spoke on templates - why you should consider writing them, how to write them, something like that.

Dave

Add a comment