Clarion Magazine

Clarion News

- » SuitePro 2009 40% Discount
- » Noyantis CommandBars 1.17
- » Noyantis TaskPanel 1.07
- » Noyantis PropertyGrid 1.05
- » Noyantis ShortcurBar 1.18
- » Noyantis DockingPane 1.02
- » Noyantis CommandBars 1.16
- » Noyantis CalendarPro 1.18
- » SetupBuilder 6.9 Build 2516
- » SetupBuilder 6.9 Build 2509
- » Multi-Format Icons Article
- » Clarion Evangelist Folds Tent
- » gQ Low-Cost Query/Sort Builder 2.0
- » Noyantis DockingPanes 1.01
- » HCUG Meeting
- » Chicago CUG Meeting
- » Keystone 3rd Party Tools For C7
- » Noyantis TaskPanel 1.05
- » Crisp Letter Icons
- » Noyantis CommandBars 1.14
- » Clarion QuickBooks Connections Beta 0.91

[More news]

- » Clarion.NET FAQ
- » Clarion# Language Comparison

[More Clarion & .NET]

[More Clarion 101]

Latest Free Content

- » Source Code Library 2009.01.31 Available
- » Tips Vol 5 Is Now In Print!
- » A Brief History of Clarion

[More free articles]

Clarion Sites

Clarion Blogs

Latest Subscriber Content

Source Code Library 2009.01.31 Available

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

Posted Monday, February 09, 2009

Using Multiple Clarion Versions In C7: Latest Developments

There have been two C7 builds since Steve Parker wrote Using Multiple Clarion Versions In C7. With those releases, especially the most recent, there have been some changes which should be seriously welcomed by anyone wishing to use C7 to maintain applications in previous versions of Clarion.

Posted Tuesday, February 10, 2009

Understanding Clarion VIEWs, Part 1

Despite being in the language for more than 10 years now, VIEWs remain largely unknown to most Clarion programmers. Bruce Johnson shows how easy it is to define and use views. Part 1 of 2.

Posted Wednesday, February 11, 2009

Understanding Clarion VIEWs, Part 2

Having showed in Part 1 how easy it is to define and use views, Bruce Johnson covers the key features of view orders and filters. Part 2 of 2.

Posted Wednesday, February 11, 2009

Tips Vol 5 Is Now In Print!

Clarion Tips & Techniques Volume 5 is now in print, and normal pricing applies. Volume 5 is packed with choice articles from the pages of Clarion Magazine, covering topics from the Clarion language to the ABC class library, from the Windows API to the template language to SQL. There's even a section on Clarion#! **Read more...**

Purchase now!

Clarion Tips & Techniques
Volume 5
Time-saving, money-saving, sanity-saving gems from the pages of Clarion Magazine.

Posted Thursday, February 12, 2009

Clarion 7 Beta 5 Released

SoftVelocity has released a fifth C7 appgen beta. This release has by far the longest list of bug fixes and sports some important improvements including a new template editor. Read Bob Z's blog entry, and stay tuned to ClarionMag for more on this beta.

Posted Monday, February 16, 2009

Migrating From MySQL To PostgreSQL, Part 1

The great ClarionMag migration from MySQL to PostgreSQL is under way! Dave Harms explains the rationale behind the move and covers some of the schema conversion issues.

Posted Friday, February 20, 2009

Migrating From MySQL To PostgreSQL, Part 2

The great ClarionMag migration from MySQL to PostgreSQL is under way! In this second of two parts Dave Harms looks at some of the problems that arose during the data transfer.

Posted Friday, February 20, 2009

C7 AppGen Build 4869 (updated)

ClarionMag takes a look at C7 AppGen beta build 4869. Important features include the new template editor and embeditor error handling. Updated Feb 25 with additional TXD information.

Posted Wednesday, February 25, 2009

Building C7 Apps With MSBuild

C7, like Clarion#, uses MSBuild to manage the compile process. But MSBuild can do much more, as Dave Harms explains.

Posted Saturday, February 28, 2009

A Brief History of Clarion

A recent discussion of DevCons past prompted Dave Harms to compile an incomplete history of Clarion. Help fill in the blanks!

Posted Saturday, February 28, 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

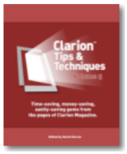
Printed Books & E-Books

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
Time-saving, money-saving, sanity-saving gems from the pages of Clarion Magazine.
Edited by David Harms

- » 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

SuitePro 2009 40% Discount

MotleySoft has a 40% discount available for Clarion developers who want to buy SuitePro 2009 13.x.x, with 30 days or 1 year support. Use the coupon code CLARION and press the Update/Recalculate button. You will see a new discounted price (and small (E) icon).

Posted Wednesday, February 25, 2009

Noyantis CommandBars 1.17

Version 1.17 of the Noyantis CommandBars wrapper template has been released. Modifications include: Width calculation incorrect when other Noyantis Wrapper templates present. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 24, 2009

Noyantis TaskPanel 1.07

Version 1.07 of the Noyantis TaskPanel wrapper template has been released. Modifications include: Codejock v13.0.0 compatibility added; Height calculation enhanced during window resize; Bug fix - Control definition incorrect for Legacy apps compiled in multi-DLL mode. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 24, 2009

Noyantis PropertyGrid 1.05

Version 1.05 of the Noyantis PropertyGrid wrapper template has been released. Modifications include: Codejock v13.0.0 compatibility added; Spin Controls added; Checkbox Controls added; Radio Options added; Checkbox Options added; New method added - "AddItemSpin"; New method added - "AddItemCheckbox"; New method added - "AddItemRadioOption"; New method added - "AddItemCheckOption"; Bug fix - Control definition incorrect for Legacy apps compiled in multi-DLL mode. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 24, 2009

Noyantis ShortcurBar 1.18

Version 1.17 of the Noyantis ShortcutBar wrapper template has been released. Modifications include: Codejock v13.0.0 compatibility added; Height calculation enhanced during window resize; Reading and hiding of System MenuBar now handles bars on both standard windows and application frames; Bug fix - Control definition incorrect for Legacy apps compiled in multi-DLL mode. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 24, 2009

Noyantis DockingPane 1.02

Version 1.02 of the Noyantis DockingPane wrapper template has been released. Modifications include: Codejock v13.0.0 compatibility added; Bug fix - Control definition incorrect for Legacy apps compiled in multi-DLL mode. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 24, 2009

Noyantis CommandBars 1.16

Version 1.16 of the Noyantis CommandBars wrapper template has been released. Modifications include: Codejock v13.0.0 compatibility added; Height calculation enhanced for application frame; Width calculation enhanced during window resize; Reading and hiding of System MenuBar now handles bars on both standard windows and application frames; Bug fix - Status Bar hidden for Designer Bars; Bug fix - Frame Extension: Popups were still being displayed when the control had lost focus.(code accidently omitted during v1.14); Bug fix - Control definition incorrect for Legacy apps compiled in multi-DLL mode. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 24, 2009

Noyantis CalendarPro 1.18

Version 1.18 of the Noyantis CalendarPro wrapper template has been released. Modifications include: Codejock v13.0.0 compatibility added; TimeLine view added; New Event, Procedure "CellSelected_????" and embed point added to inform the user when a different calendar cell has been selected; New Procedure "NoEventSelected_????" added; Right Click options enhanced and better C55 compatibility added; v13.0.0 of the Control changed the "Week (Detailed)" default view. To correct this 2 new methods have been added:-; New method "GetWeekDetailMode"; New method "SetWeekDetailMode"; Bug fix - Control definition incorrect for Legacy apps compiled in multi-DLL mode; Bug fix - DatePicker control could initialize too early under certain circumstances. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 24, 2009

SetupBuilder 6.9 Build 2516

Lindersoft has released SetupBuilder 6.9 Build 2516. This release is available, free of charge, to all SetupBuilder customers who have an active SetupBuilder maintenance subscription plan. If you do not have an active subscription plan, please contact your account manager at sales@lindersoft.com.

Posted Tuesday, February 24, 2009

SetupBuilder 6.9 Build 2509

Lindersoft has released SetupBuilder 6.9 Build 2509. This is an update to the latest stable release, and contains some important bug fixes and improvements. This build also brings SetupBuilder 6 in-sync with the latest SetupBuilder 7 beta build. This release is available, free of charge, to all SetupBuilder customers who have an active SetupBuilder maintenance subscription plan. If you do not have an active subscription plan, please contact your account manager at sales@lindersoft.com.

Posted Tuesday, February 24, 2009

Multi-Format Icons Article

Dave Beggs has posted an article about how to use multi-format icons in Clarion.

Posted Tuesday, February 24, 2009

Clarion Evangelist Folds Tent

Stu Andrews has resigned his post as Clarion Tech Evangelist. He will, however, continue with the podcast and remains an ardent Clarion fan.

Posted Tuesday, February 24, 2009

gQ Low-Cost Query/Sort Builder 2.0

gQ - Query/Sort Builder version 2.0 is now available, and special pricing is in effect until Feb 20, 2009. The Full Version includes all minor version upgrades and email support for $59 (will be $79 after Feb 20th). The "Want It Cheap!" Version is exactly the same version as above, also includes minor version upgrades, but does not include support - you're on your own (of course, you have the source to modify to your needs) - for $24 (will be $30 after Feb 10th). This price applies to both new licenses and upgrades.

Posted Tuesday, February 10, 2009

Noyantis DockingPanes 1.01

Version 1.01 of the Noyantis DockingPanes wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; Legacy compatibility added with Example app. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, February 10, 2009

HCUG Meeting

The next HCUG meeting is at the Venu Map Trustco office, Unit 304 Oakmont Building, Somerset Links Office Park,(Off De Beers Ave) on February 11, 2009 (Wednesday) from 17:30 for 18:00. Duration is around 2-3 hours. Bring a cool drink or beer of your preference. Cover charge is R30 per person cash (for grub, coffee, tea etc). Topics: Continue from where we left off last; Q&A; Some surprise topics. Please R.S.V.P. to rsvp at clarionusergroups.com

Posted Tuesday, February 10, 2009

Chicago CUG Meeting

The next meeting of the Chicago CUG will be held Wednesday February 11th, 2009 at 6:30 pm in suite 270, in the Nebo Systems offices. The address is 1 South 450 Summit Ave. Suite 270,Oakbrook Terrace, IL 60181. The sign at the front of the office complex says Summit Oaks. Steve Parker will be giving the presentation on Clarion 7, showingthe latest build with the AppGen.

Posted Tuesday, February 03, 2009

Keystone 3rd Party Tools For C7

Keystone's 3rd party tools for Clarion Release 7.0 are now available. Registered owners can download from the web site. You will need to contact support to confirm your email address and obtain new installation keys.

Posted Tuesday, February 03, 2009

Noyantis TaskPanel 1.05

Version 1.05 of the Noyantis TaskPanel wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example

app; Legacy compatibility added with Example app. The new version can be downloaded from the Members area using the original download and registration details contained in sales emails.
Posted Tuesday, February 03, 2009

Crisp Letter Icons

D-Icons has released its new Crisp Letter icon set. This is a set of letter and symbol icons based on a clean fresh pastel theme, to complement the d-crisp Icons set released a couple of weeks ago. There are 93 different designs, all in 12 vibrant colors at sizes from 128x128 down to 16x16. The symbols include the alphabet in upper and lower case, numbers, and most symbols available on a standard keyboard. The colors exactly match the d-crisp Icons.
Posted Tuesday, February 03, 2009

Noyantis CommandBars 1.14

Version 1.14 of the Noyantis CommandBars wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app (C55 Frame Extension to be completed); Legacy compatibility added with Example app; New methods added - 'SetComboSelection' & 'GetComboSelection'; Keybindings on Designed Bars could cause app to GPF if pressed. The new version can be downloaded from the Members area using the original download and registration details contained in sales emails.
Posted Tuesday, February 03, 2009

Clarion QuickBooks Connections Beta 0.91

Clarion QuickBooks Connections is a set of classes and templates that ease the access and update of QuickBooks data. The intro price is $149.
Posted Tuesday, February 03, 2009

SetupBuilder 6.9 Build 2482 Developer Edition Hotfix

This hotfix addresses a "#copy file..." compiler directive issue. To get the latest product version, select Help | Check for Updates... in your SetupBuilder IDE.
Posted Tuesday, January 20, 2009

Base64 Example

John Griffiths adapted some GPL (General Public License) C++ source for Base64 encoding and has incorporated it into a CW6.3 APP.
Posted Tuesday, January 20, 2009

SetupBuilder 6.9 Build 2481

Lindersoft has released SetupBuilder Version 6.9 Build 2481, the latest edition of its award-winning Installation Authoring and Configuration Management system for Microsoft Windows based applications. SetupBuilder Version 6.9 deploys applications to the Microsoft Windows Platform, including Windows 7, Windows Vista, Windows Server 2008 and Windows x64. This release is available, free of charge, to all SetupBuilder customers who have an active SetupBuilder maintenance subscription plan. This build brings SetupBuilder 6 in sync with SetupBuilder 7.
Posted Tuesday, January 20, 2009

Noyantis ShortcutBar 1.15

Version 1.15 of the Noyantis ShortcutBar wrapper template has been released. Modifications include: Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; BUG FIX: Legacy detection enhanced. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, January 20, 2009

J-HTML Improved

A new "Advanced" tab has been added to the J-Html control. At the moment this tab enables you to easily override hyperlink targets, and to set whether or not users can drop documents into the control. J-Html contains a huge amount of code that has never really been documented or made public. Typically this is code written for specific clients, or for internal use. The next version of J-Html will open up some of these features through this new "Advanced" tab.

Posted Tuesday, January 20, 2009

Noyantis PropertyGrid 1.03

Version 1.03 of the Noyantis PropertyGrid wrapper template has been released. Modifications include: Codejock ActiveX Auto Registration facility added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, January 20, 2009

Noyantis CalendarPro 1.14

Version 1.14 of the Noyantis CalendarPro wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; Legacy compatibility added with Example app; New methods added - GetActiveStartDate & GetActiveEndDate; 2 new parameters added to NewDate_???? procedure (Active Start and End Date); New method added - SelectDayTime; New methods added - SetDayStartTime & SetDayEndTime; Restrict parameter added to SetDayLimits method; User definable conditions added to Right Click options; Override Office 2007 Theme facility added (includes over 15 new methods for altering appearance); Additional Embed points added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, January 20, 2009

Noyantis ShortcutBar 1.15

Version 1.15 of the Noyantis ShortcutBar wrapper template has been released. Modifications include: Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; BUG FIX: Legacy detection enhanced. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, January 20, 2009

Noyantis CalendarPro 1.14

Version 1.14 of the Noyantis CalendarPro wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; Legacy compatibility added with Example app; New methods added - GetActiveStartDate & GetActiveEndDate; 2 new parameters added to NewDate_???? procedure (Active Start and End Date); New method added - SelectDayTime; New methods added - SetDayStartTime & SetDayEndTime; Restrict parameter added to SetDayLimits method; User definable conditions added to Right Click options; Override Office 2007 Theme facility added (includes over 15 new methods for altering appearance); Additional Embed points added. The new version can be

downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, January 20, 2009

Clarion Magazine

A Brief History of Clarion

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.
1982 (approx)	Work begins on Clarion 1.0	Team members include Monty Shaw, Bob Steinke, Gary Liming, John Herron
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.
1985	Borland moves to London	Begins developing new compiler line for Modula-2, C and C ++
1986	Clarion 1.0 released, April.	Dongle almost kills the product, removed in 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 Hotel in Ft. Lauderdale. CDD 3.0 announced, but little shown of substance. Keynote speakers included Bruce Barrington, Will Fastie, and Niels Jensen.
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.
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.
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 (?)
1994?	DevCon West	Year not confirmed, and there were at least two of these. Stamos says one was at the Hilton, the other possibly at the Hyatt. John Griffiths recalls the DoubleTree hotel and Bruce Barrington showing an early CW beta, presumably either of 1.5 or 2.0, depending on the year.
1994	Aussie DevCon	Sydney, Australia
1994	Clarion Software becomes Topspeed Corporation	
1995	**DevCon #5?**	Location unknown. In fact, whether there was a DevCon in '95 is unknown.
1995?	South American DevCon?	Argentina?
1995	CW 1.5 released, September	
1995	CDD 3.1 released	
1996	Final CW 1.5 release, January	Version 1.502. Includes support for 32 bit executables.
1996	Kerrville gathering	Kerrville, TX. Pics. Ross Santos discovers fire ants.
1996	CW 2.0 Released, June	Adds object orientation, ActiveX support, enhanced SQL support

1996	**DevCon #6**	Held at the Orland airport, and generally regarded as the absolute worst DevCon location.
1996	Arizona DevCon, Nov	Hosted by Kathryn & Phil Carroll, and I think Larry Teames had just a bit to do with it as well
1997	Aussie DevCon, Feb	Sydney, Australia
1997	Final CW 2.0 release, March	
1997	Euro DevCon, April	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	**DevCon #7**, Sept	Radison Hotel, Ft. Lauderdale. Keynote by Peter Coffee.
1997	Clarion 4 released, Dec	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	Sydney DevCon, Feb	Sydney, Australia. The President's Hotel in King's Cross.
1998	eData is formed	Hank Asher goes on to hire most of the London team.
1998	Clarion 5.0 released, May	First appearance of Professional and Enterprise versions
1998	Euro DevCon	May, Ramada Gatwick Airport
1998	**DevCon #8**, Sept	Marina Marriott hotel in Ft. Lauderdale. Keynote: Mr. Edward E. Iacobucci - Citrix chairman, chief technical officer and vice president of strategy and technology.
1998	ETC I, May	Lee White holds the first East Tennessee Clarion Conference in Gatlinburg.
1998	ConVic, Nov	Ballarat, Victoria, Australia. The first of Chris Livingstone's ConVic conferences
1998	South African DevCon, Nov	Johannesburg, Indaba Hotel, Fourways
1999	ClarionMag begins publication, Feb	
1999	Aussie DevCon, March	Sydney, Australia. Article.
1999	Euro DevCon, April	Amsterdam, Golden Tulip Barbizon Palace

1999	DevCon Latin America, May	Article
1999	Roy Rafalco resigns, Frank Watts hired as new CEO	
1999	ConVic '99	Mt. Buffalo, Australia. Article. Article.
1999	Clarion 5.0B released, June	Final 5.0 release
1999	Clarion 5.5 beta	
1999	**DevCon #9**, Sept	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.
2000	ETC II	Lee White holds the second East Tennessee Clarion Conference in Gatlinburg.
2000	ConVic	Daylesford, Australia. Article.
2000	Clarion 5.5 released, August	Adds interfaces to the language
2000	South American ConDev	Sao Paulo, Brazil?
2001	South American ConDev, October	Argentina (last Argentinian Clarion conference before the economic collapse)
2001	ConVic 2001, March	Mt. Buffalo, Australia. Article.
2002	Clarin 5.507 released, January	Final 5.5 release
2002	South American ConDev	??
2002	ConVic 2002	Geelong, Australia. Article.
2002	ETC III	Lee White holds the third East Tennessee Clarion Conference in Gatlinburg. Articles.
2003	ConVic 2003	Geelong, Australia. Article.
2004	ETC IV	Lee White holds the fourth East Tennessee Clarion Conference in Gatlinburg. Articles.
2004	South American ConDev	Porto Alegre.
2004	**DevCon #10**, Orlando	Held at the Disney contemporary resort. First SoftVelocity DevCon. Some Clarion# vaporware shown. Articles.
2005	South American ConDev	Foz do Iquaco, Brazil. Article #1.Article #2.

2006	Aussie DevCon, May	Geelong, Australia. Articles.
2007	First Clarion 7 hand coder's release	Build 1457 was released on Feb 23, 2007
2007	Aussie DevCon, May	Sydney, Australia. Articles.
2007	First Clarion# hand coder's release	Nov 17
2008	Aussie DevCon, May	Eden, Australia. Articles. First public appearance of AppGen.
2008	C7 AppGen Third Party Release	Oct 21
2008	C7 AppGen CSP release	Dec 24

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.

Add a comment

Clarion Magazine

Using Multiple Clarion Versions In C7: Latest Developments

by Steven Parker

Published 2009-02-10

There have been two C7 builds since I wrote Using Multiple Clarion Versions In C7. With those releases, *especially* the most recent, there have been some changes. These changes should be seriously welcomed by anyone wishing to use C7 to maintain applications in previous versions of Clarion.

Build 4749, the third public AppGen build, and the build current at the time of this writing, includes the following paragraph at the beginning of the release notes:

> FEATURE: The IDE now notifies you when you open a solution using a different Clarion version than is currently selected. You can choose to open in the version the solution was last opened in or the currently selected version.

This feature makes using C7 to maintain apps created in previous versions of Clarion much, much easier. Indeed, using C7 as a general Clarion development shell becomes almost "falling off a log easy." (Build 4790 is current at the time of publication but nothing of significance has changed regarding building apps with multiple versions of Clarion.)

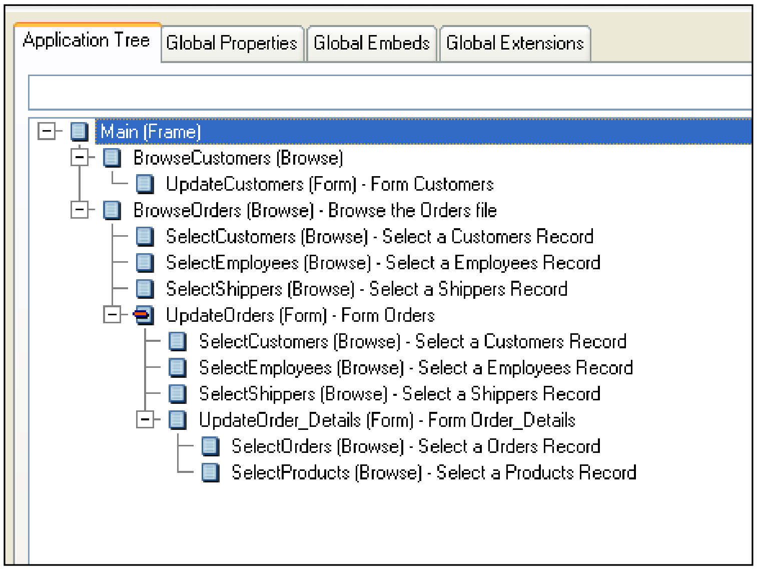
In fact, the major problems in using C7 as a "universal Clarion shell" are:

- Neither .APP nor .TXA files are backwardly compatible
- "Older" templates sometimes do not parse correctly and, therefore, fail to register.

TXA compatibility

Because the .APP file format has, of necessity, changed, if there is any hope of being able to move back to earlier versions, should that become necessary, the TXAs created by C7 must be readable by the original version.

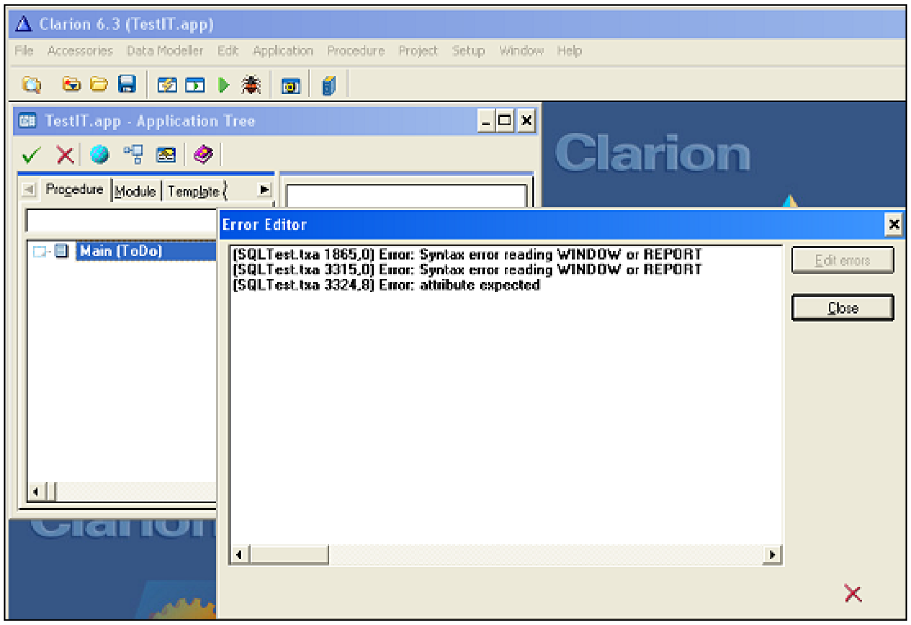
To test this, I exported a C7 app to TXA.



```
Main (Frame)
    BrowseCustomers (Browse)
        UpdateCustomers (Form) - Form Customers
    BrowseOrders (Browse) - Browse the Orders file
        SelectCustomers (Browse) - Select a Customers Record
        SelectEmployees (Browse) - Select a Employees Record
        SelectShippers (Browse) - Select a Shippers Record
        UpdateOrders (Form) - Form Orders
            SelectCustomers (Browse) - Select a Customers Record
            SelectEmployees (Browse) - Select a Employees Record
            SelectShippers (Browse) - Select a Shippers Record
            UpdateOrder_Details (Form) - Form Order_Details
                SelectOrders (Browse) - Select a Orders Record
                SelectProducts (Browse) - Select a Products Record
```

Figure 1. Original, C7, app tree

I then created a new C6 app, in C6, and imported the TXA. It errored:


Clarion 6.3 (TestIT.app)
File Accessories Data Modeller Edit Application Procedure Project Setup Window Help
TestIT.app - Application Tree
Procedure Module Template
Main (ToDo)
Clarion
Error Editor
Edit errors
Close

```
[SQLTest.txa 1865,0] Error: Syntax error reading WINDOW or REPORT
[SQLTest.txa 3315,0] Error: Syntax error reading WINDOW or REPORT
[SQLTest.txa 3324,8] Error: attribute expected
```

Figure 2. TXA import errors

The "Edit Errors" button didn't take me to any source. But the import must have proceeded and then stopped at the point of the error because the C6 app tree was not empty. The application tree, however, was incomplete:

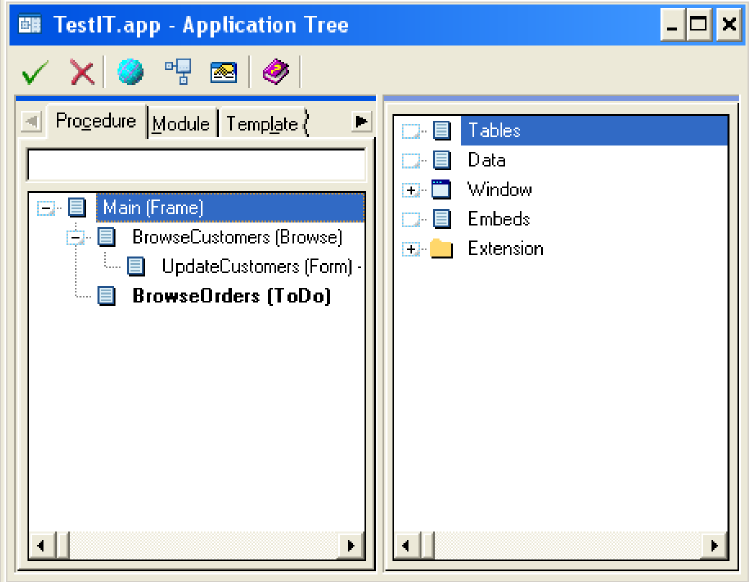


Figure 3. Resulting C6 .APP tree

So, unless Softvelocity changes the TXA format again or provides a TXA conversion tool, be sure in your own mind before committing to C7 to maintain C6 apps And even if such a tool appears, remember that C7 dictionaries now export to an XML format rather than TXD!

Note: Richard Rose has created a tool which will convert C7 TXAs into C6 TXAs. He says that it is a work in progress. Check on the news groups; Mr. Rose has offered to send it to whoever needs it.

Templates

Many templates will require fixing. The required fixes are described in the on-line help. Also see the previous article for more details.

My own experience is that Softvelocity's C6 (9056) templates, Mike Hanson's SuperStuff, Lee White's RPM and Capesoft's templates require no updating. Each registered in C7 without error or warning.

I do have some third party C6 templates that showed problems. But I am not going to point out the vendors as (1) my copies may be older and newer versions may register in C7 properly or (2) these templates may have been updated by their authors since the C7 AppGen was introduced. If you have issues that you cannot easily fix, contact the vendor.

The most common problems are missing END statements (particularly in Control templates) and misplaced or unexpected SEQ statements. These *are* easily fixed; just add the END or delete the SEQ statement.

I also experienced a number of problems with Clarion 5.5 templates, including several Softvelocity OEM templates. I could not fix all of them. (As I recall, the 5.5 abmail.tpw file had problems I did not immediately see how to fix. But, I do not believe I use this template in the few 5.5 apps I still have to maintain. So, I haven't pursued this.)

Now, back to our show

The big change, the change underlying and giving substance to the release note quoted at the beginning of this article, is in the "preference file," for lack of a better term.

If you look in the %appdata%\SoftVelocity\Clarion\7.0\preferences directory, you will find a number of files. All are XML files and there will be at least two for each .APP (sometimes more; I haven't figured out how or why additional file pairs are created but I do know that I have deleted them and let the IDE recreate without incident).

For example, I have a small app in which I enter part of a UPC code (6, 10 or 11 digits). The program computes the check digit and full UPC code. It is called CalcCheckDigit.APP. In the "preferences" directory, I find:

CalcCheckDigit.**cwproj**.2dd5b780.xml

and

CalcCheckDigit.**sln**.a7d8adc8.xml

One file, clearly, is related to the Clarion project. The other, obviously, is related to the "solution," the sln file (I have bold faced the relevant parts of the file names).

CalcCheckDigit.**sln**.a7d8adc8.xml contains the following data:

```
<Properties>
  <ActiveVersion value="Current" />
  <ActiveNetVersion />
  <StartupProject value="" />
  <ActiveConfiguration value="Debug" />
  <ActivePlatform value="Win32" />
  <ProjectBrowserState value="[]" />
</Properties>
```

Lines like

```
<ActiveConfiguration value="Debug" />
```

and

```
<ActivePlatform value="Win32" />
```

seem pretty clear. It is the

```
<ActiveVersion value="Current" />
```

line that is important. If the .APP had been opened in Clarion 6 (Build | Set Clarion Version from the IDE main menu), it would read as follows:

```
<ActiveVersion value="Clarion 6.0 Enterprise Edition" />
```

But, in the previous AppGen builds, this line reads:

```
<ActiveVersion value="" />
```

if the application had been created in or last updated to C7.

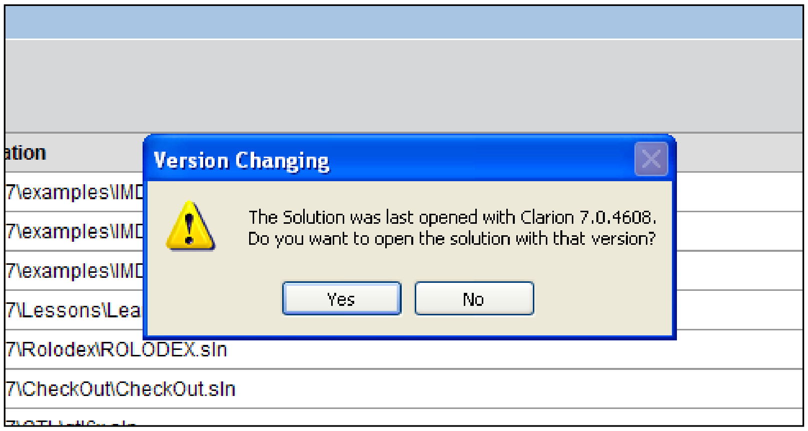
Because there was no value in the ActiveVersion node, a C7 app could easily "revert" to C6 if the previous app I opened had been a 6.x app. This is why, in the last article, I was so adamant about (1) closing the .APP *and* (2) closing the solution, two separate steps, before (3) setting/resetting the Clarion version to use.

That is, if ActiveVersion value="", whatever version of Clarion had last been set in the IDE or used to open an application is used for the application currently being opened. This is important because the IDE needs to close the template registry for the current Clarion version and read/load the registry for the desired version if the target version of Clarion is different.

Now, from build 4749, the IDE will warn me and ask what I want to do. For example, some (but not all) of the apps I opened in 4608 wrote

```
ActiveVersion value="Clarion 7.0.4608" />
```

to the preference file (I never did figure out why sometimes this was written and sometimes not). In the previous release of C7 (4725), I got a message about 4608 (4725's predecessor) not being a recognized release of Clarion. Now, I get the more meaningful message in Figure 4.



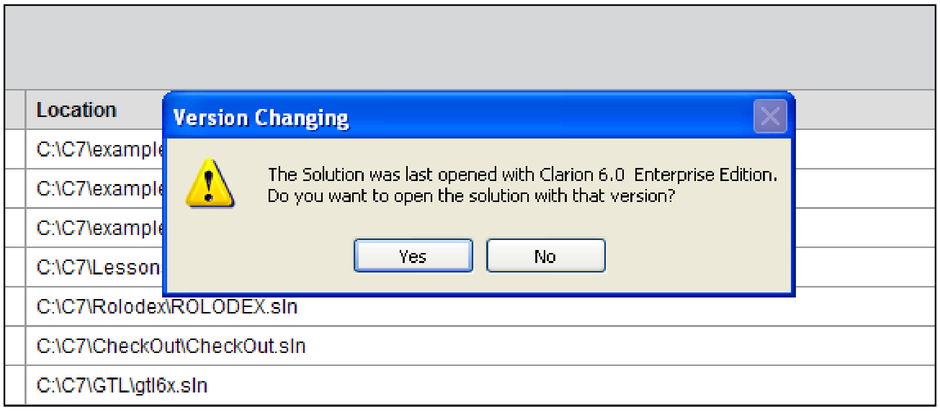
Version Changing

The Solution was last opened with Clarion 7.0.4608.
Do you want to open the solution with that version?

Figure 4. Message if last opened in 4608

If I click Ok, which I didn't as I'd uninstalled 4608, the IDE would try to load that version. If I click No, the current iteration of Clarion will be used.

If the next app I open is one I last opened in 6, I get the message in Figure 5.



Location
C:\C7\example
C:\C7\example
C:\C7\example
C:\C7\Lesson
C:\C7\Rolodex\ROLODEX.sln
C:\C7\CheckOut\CheckOut.sln
C:\C7\GTL\gtl6x.sln

Version Changing

The Solution was last opened with Clarion 6.0 Enterprise Edition.
Do you want to open the solution with that version?

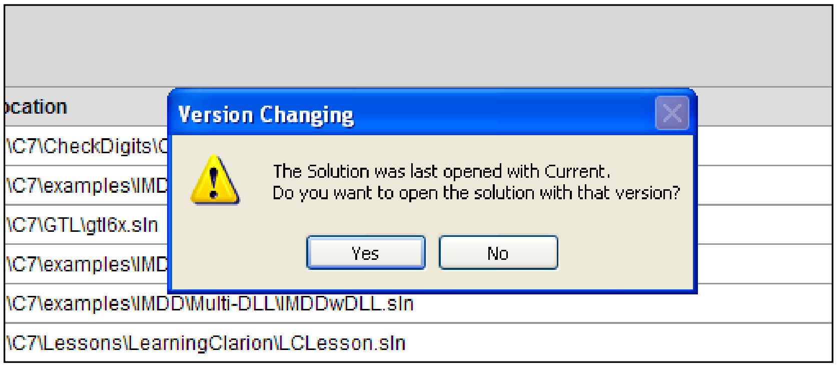
Figure 5. Message if last opened in Clarion 6.x

If the app I am opening is in the same Clarion release as the last app opened, I get no message.

This is now much more developer-friendly.

Among other things, this seems to provide a mechanism for "upgrading" apps from earlier Clarions to C7. "When I'm ready," of course.

It also provides a reverse migration path. Close all .APPs and solutions. Change the Clarion version from the IDE (Build | Set Clarion Version). When I open an app that had last been opened in Clarion 7, I get the warning window in Figure 6.



ocation
\C7\CheckDigits\C
\C7\examples\IMD
\C7\GTL\gtl6x.sln
\C7\examples\IMD
\C7\examples\IMDD\Multi-DLL\IMDDwDLL.sln
\C7\Lessons\LearningClarion\LCLesson.sln

Version Changing

The Solution was last opened with Current.
Do you want to open the solution with that version?

Figure 6. Reverting to a prior version

I *can* elect to open the app in the older version.

By the way, the IDE "remembers" the last set version between sessions. If I close C7 while working on a C6 or 5.5 app or project, when I reopen C7, that version is still "current."

But wait! There's more!

The ActiveVersion entry in the preference file was not always written in the previous releases of C7 AppGen. Realizing the importance of the ActiveVersion entry in the preference file, I manually updated it to keep me on track. (If I didn't, as mentioned earlier, I could - I did - accidentally change the active Clarion version.)

As of Build 4749, ActiveVersion is written whenever an .APP is touched (I don't see the value until I close the .APP but even if I cancel out of the .APP, the expected value ends up in the preference file). So, knowing that "Clarion 6.0 Enterprise Edition" had been written for my 6.x apps, safeguarding my C7 apps seemed simple.

I took the expedient of ensuring that C7 was set as the current version by opening and closing each .APP I knew I wanted to be C7. Checking each preference file, I saw the appropriate, expected values in ActiveVersion. (Accidentally picking a C6 app is no problem, the IDE prompts me.)

The proof of the pudding

After performing this exercise, I opened an .APP I knew to be a C6 app (GTL, my C6 batch compiler). Examining the IDE title bar (Figure 7) I could see that the version *had* changed.

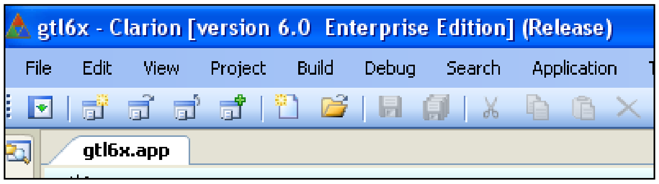


Figure 7. Title bar information, C6 .APP

Note that the C7 title bar tells me not only the .APP name, but the Clarion version and build mode (debug or release) also. Then, opening a known C7 app results in the display shown in Figure 8.

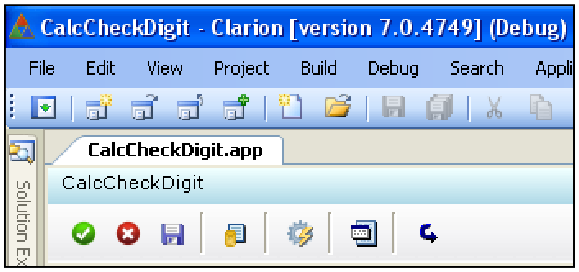


Figure 8. Title bar information, changing to a C7 .APP

I, again, see that the Clarion version has been set correctly.

By the way, between opening GTL and CalcCheckDigit, I violated my own rule. I closed the .APP but I didn't close the solution....

Caveat

Since the version information is stored in the preferences file, deleting the preferences file or copying the solution to a different computer will result in the target version information being lost. You will have to reset the Clarion version explicitly from the IDE main menu. This also means some care should be taken when an APP resides in several different solutions, since the potential exists for generating/compiling the same application for different releases of Clarion.

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

Understanding Clarion VIEWs, Part 1

by Bruce Johnson

Published 2009-02-11

The Clarion FILE structure is a cornerstone of the Clarion language, and has been largely unchanged since the very early Clarion for DOS days. As such it is well understood, and most programmers use file handling in embed code all the time.

The Clarion VIEW structure slid onto the scene with Clarion for Windows 1.5, and was immediately used extensively by the templates. However despite being in the language for more than 10 years now, views remain largely unknown to most programmers.

In essence a view is a virtual file; in other words, it's a structure that acts as like a table but isn't declared in the dictionary. It's able to include fields from more than one table, as well as sort and filter the records being included.

> **Note:** A quick word for those familiar with SQL. If you are using SQL you can define server-side views. However this article is about client-side views, which are declared in your Clarion program regardless of the backend being used.

Portability

Part of the reason for programmers ignoring views is that FILE structures are remarkably efficient in an ISAM environment (TPS files being the most common example of an ISAM database). When programmers move to SQL backends the inefficiencies of FILEs became obvious, and the usual approach to improve matters is to make use of the PROP:SQL command. Unfortunately though the PROP:SQL command is not portable to ISAM file drivers, and any specific PROP:SQL command may be limited to a specific SQL server version. PROP:SQL offers a gain in performance over FILE, but at the expense of portability.

Why should this matter? Portability is seldom a requirement specified at the beginning of a project. And by the time it becomes an issue it is often too late to do anything about it without incurring a substantial amount of extra work. Creating portable applications will never be completely transparent, but by leveraging portable code from the very beginning you can make your life easier, and your work load lower in the long run.

Writing portable code is an act of will; it doesn't happen by accident. Getting into the habit of writing portable code is a learned skill. And being able to write code using client-side VIEW structures is a key part of growing as a programmer.

Efficiency

A VIEW structure cannot make ISAM file access more efficient (although it can result in smaller, neater code.) ISAM is what it is, and even today it is a remarkably powerful solution. While SQL offers a number of advantages, it comes at a financial, resource and human cost.

However, coding with views is no less efficient for ISAM, but at the same time makes your code vastly more efficient when moving to SQL. Even if one specific project never moves to SQL, it is nevertheless a good habit to get used to coding views.

For SQL, client-side views are dramatically more efficient than accessing tables directly, in all but a small number of situations. By specifying only the fields you actually need and by supporting server-side filtering and ordering, the view is able to leverage features offered by the server when these are available.

Programs that are more efficient are also faster, and this improved efficiency translates into raw program performance. Since views lighten the load on the server they improve not just your program, but all programs accessing the server at the same time.If you are using a SQL backend views result in less network traffic, which means the network is more efficient as well.

Flexibility

Basic file structures are really easy to code when you are dealing with a single file. However, adding secondary loops and other file lookups increases the size of your code. By contrast views are designed to make secondary files and lookups easy to deal with.

Functionality

Views can do some things that are difficult, or impossible, to do with simple file access commands. For example they can sort on secondary fields.

Coding the VIEW structure

The key to views is, not surprisingly the VIEW structure. If the structure is set up correctly, then using it, in code, is both simple and very familiar.

Here's a basic VIEW structure;

```
ThisView VIEW(invoices)
         END
```

In this case no fields are specified, so Clarion helps you out by assuming you really want all the fields in that table. A better way to declare this view might be:

```
ThisView VIEW(invoices)
           PROJECT(inv:date)
           PROJECT(inv:invoicenumber)
           PROJECT(inv:amount)
           PROJECT(inv:customerId)
         END
```

In this case only the four fields you actually need are being fetched from the table. The PROJECT statement simply tells the view which specific fields to include. It is pronounced as in "Project onto a screen" not "I am starting a new project".

Of course what you really want is not the customer id, but the customer name. That's stored though in the customers file, not the invoices file. The VIEW allows you to join fields together even if they are in different tables.

```
ThisView VIEW(invoices)
           PROJECT(inv:date)
           PROJECT(inv:invoicenumber)
           PROJECT(inv:amount)
           JOIN(cus:idkey,inv:customerid)
             PROJECT(cus:customerName)
```

```
     END
    END
```

At first you'll probably need to check the help to remind yourself how the JOIN is constructed, but after a short time it's really easy to recall. Just remember the first parameter in the JOIN specifies (directly or indirectly) the secondary table being joined to.

By default the view returns a record for each record in the primary table, even if the secondary table has no record. In the example above every invoice is returned even if the inv:customerid field does not match any customer. This default JOIN behavior is known as a *left outer join*, and is the most common kind of join. Ordinary related file lookups in browses function this way.

However if you add INNER at the end of the JOIN statement like this:

```
JOIN(cus:idkey,inv:customerid),INNER
```

then you have an *inner* join: parents with no children are automatically ignored, and only invoices with valid customers will be returned.

In the above view example the customer record is like doing a lookup. There is only one customer for each invoice. However the join also works where there are multiple child records:

```
ThisView VIEW(invoices)
     PROJECT(inv:date)
     PROJECT(inv:invoicenumber)
     PROJECT(inv:amount)
     JOIN(lin:idkey,inv:invoicenumber)
      PROJECT(lin:quantity)
     END
    END
```

There are typically many line items for each invoice. In this case the view will contain a record for each line-item which combines the invoice information, and the line-item information.

You can mix both many-to-one and one-to-many relationships within the same view. In the VIEW structure below there are many line items to each invoice, but one product per line item, and indeed one customer per invoice.

```
ThisView VIEW(invoices)
     PROJECT(inv:date)
     PROJECT(inv:invoicenumber)
     PROJECT(inv:amount)
     JOIN(cus:idkey,inv:customerid)
      PROJECT(cus:customerName)
     END
     JOIN(lin:idkey,inv:invoicenumber)
      PROJECT(lin:quantity)
      PROJECT(lin:price)
       JOIN(Pro:IdKey,lin:ProductId)
        Project(Pro:ProductName)
       END
      END
     END
```

The view is very flexible when it comes to joining secondary files. Even though the most common use is show above, using a key and linking fields you can also specify just the secondary table name and any expression you like to make the join.

```
ThisView VIEW(invoices)
      PROJECT(inv:date)
      PROJECT(inv:invoicenumber)
      PROJECT(inv:amount)
      JOIN(Customers,'Cus:Id = inv:customerid')
        PROJECT(cus:customerName)
      END
     END
```

You can include filter and order statements in the VIEW structure itself, but I prefer to set these in the code. They are the hardest part of the VIEW structure to get correct, and they are somewhat easier to debug in the code. This however is a personal preference, so feel free to add these to the VIEW declaration if you prefer. I'll discuss filters and orders in more detail in part two.

Writing the code to process the VIEW

Looping through a view is, in many ways, very similar to looping through a table. If you are familiar with the file driver commands (used by DOS and the Clarion legacy templates) then the view access statements will be even more familiar. In a cruel irony it is the ABC-only brigade that will need to learn some new commands here.

The basic code structure follows a very simple pattern. Here's the pseudocode:

```
PushBind
OPEN all tables
Bind File Records
OPEN view
Optionally set PROP:Order
Optionally set PROP:Filter
SET View
Loop through View
  Using NEXT, PREVIOUS, SKIP and REGET.
End
CLOSE View
CLOSE all tables
PopBind
```

Binding

The PushBind and PopBind statements are advisable if you use BIND to bind variables for use in filters. PushBind sets up a new scope for BIND statements, and PopBind removes that scope and stores the previously stored state. Even if you're not using BINDing right now it's good idea to wrap the entire operation with PushBind and PopBind.

Those are the basics of using views. But in most cases you'll also want to add sorting and/or filtering. I'll discuss those options next time.

Living in Cape Town, South Africa, Bruce Johnson is a part-owner of CapeSoft and has been programming in Clarion since 1992. He authored the successful "Programming in Clarion's ABC" book and has been involved in some of Clarion's most popular accessories. When not programming he enjoys cooking, and sports - the one as a direct result of the other.

Reader Comments

Add a comment

Clarion Magazine

Understanding Clarion VIEWs, Part 2

by Bruce Johnson

Published 2009-02-11

In part one of this article I discussed the basic concepts of views and showed some simple view processing code. Now it's time to go on to two more important view concepts: filtering and ordering.

PROP:Order

The PROP:Order of the view determines the order of the records in the result set. It's up to the view engine (in the case of ISAM files) and the server (in the case of SQL files) to pick a suitable key to use to get this order. If no key is available then the result set will be sorted on the fly. Not surprisingly if your PROP:Order statement can make use of a key then the result set is generated far more quickly than if the view engine or the server has to construct the order on the fly.

> **Note:** There is a misconception that SQL does not need keys as it can order on the fly. This is true (indeed it's true for TPS files as well) but SQL also benefits from keys. If your views are slow, for SQL or TPS then it's likely that a suitable key will help it go much quicker. It is true that SQL can decide what keys it should have, however your SQL backend still needs to explicitly be set in this mode. You cannot just ignore keys and expect the SQL database to maximize performance. Also note that I'm using Clarion terminology here; what Clarion calls a key, SQL databases call an index, and you typically add these to the database with a CREATE INDEX statement. In general, you'll want SQL indexes that match your Clarion keys.

Here are a couple of PROP:Order statements:

```
ThisView{PROP:Order} = 'cus:IdNumber'
ThisView{PROP:Order} = 'cus:LastName,cus:IdNumber'
```

If you create PROP:Order carefully, to match an existing key, then you need to pay special attention to two possible field qualifiers; case sensitivity and ascending or descending order.

Case sensitivity

If your key is case insensitive (and most keys are) then your PROP:Order needs to be case insensitive as well. You can do this by including the UPPER command in the PROP:Order.

```
ThisView{PROP:Order} = 'UPPER(cus:LastName),cus:IdNumber'
```

Notice the UPPER keyword is used inside the PROP:Order value; the word UPPER is passed to the engine, not the result of any UPPER command. Even though UPPER is a Clarion command, it is translated for the backend and is processed by the SQL engine.

Ascending and descending

The second property to watch out for is whether a field in the key is ascending or descending. Again be sure to match the setting in the key if you want the key to be used. The + and - signs are used to specify the sort direction. If omitted sorting defaults to ascending (+).

```
ThisView{PROP:Order} = '-UPPER(cus:LastName),+cus:IdNumber'
```

Unique orders

It's a good idea for the fields in your PROP:Order to correspond to a unique value for each row of data in the result set. This is most easily achieved by adding the primary key field to the end of the sort order. Thus an order like this:

```
ThisView{PROP:Order} = '+UPPER(cus:LastName)'
```

is not ideal, because there may be many people who share the same last name. Extending the order to be:

```
ThisView{PROP:Order} = |
  '+UPPER(cus:LastName),+UPPER(cus:FirstName)'
```

is better because it's more likely to be a unique order. However this is clearly still not guaranteed to be unique, so adding the unique field (in this case the cus:IdNumber field) to the *end* of the PROP:Order is advisable:

```
ThisView{PROP:Order} = |
  '+UPPER(cus:LastName),+UPPER(cus:FirstName),cus:IdNumber'
```

The reason for making unique orders is twofold. Firstly it means that the records will always appear in the view in the same order, thus being far less confusing to both the user, and the programmer. It also makes a number of extra view functions possible (functions like POSITION and RESET for example.)

For best results make sure all the sort order fields are PROJECTed in the VIEW structure itself.

I should point out at this point that adding additional fields to the sort order, beyond those offered by the key, usually has minimal impact on performance.

Filter versus range

In a moment I'll describe the PROP:Filter in some detail. But what about the range? Good Clarion programmers know that entering a range is critical for extracting maximum performance from a browse. So it stands to reason that the view should also have the range set.

The answer here is that there's no such thing as a range. It's a hang-over term from Clarion for DOS. The fields you enter for the range, in the templates, are simply used as part of constructing the filter. Construct the filter correctly, taking into account both the range fields and the filter fields, and you'll get maximum performance.

PROP:Filter

The most powerful part of the view engine is the PROP:Filter statement. You'll quickly discover that a well formed filter is the key to a successful view. Get the filter wrong and you may get no filtering, or possibly very slow filtering.

I'll start with a simple filter and make it progressively more complicated.

```
Thisview{PROP:Filter} = 'cus:SalesmanId = 4'
```

In this example a simple file field is matched to a static value. Here's another example, but this time the static value is a string:

```
Thisview{PROP:Filter} = 'cus:LastName = <39>Johnson<39>'
```

Notice that the string value is surrounded by quotes. There are two ways to represent a single quote as a character in a Clarion string. The first is by using the ASCII value (which is 39), the second is by using two single quote characters next to each other ('cus:LastName = "Johnson"').

I use both techniques in my own code, but for the sake of clarity, in this article I'll use the <39> method.

Remember that, if you have a key in mind, the case sensitivity of the order needs to match the key. In the same way the case sensitivity of the filter should also match the key. The filter above is case sensitive. If you'd like the string comparison to be case in-sensitive, then you use the UPPER keyword;

```
Thisview{PROP:Filter} = |
  'UPPER(cus:LastName) = <39>JOHNSON<39>'
```

So far so good. However the filter is seldom likely to be hard-coded like this. Typically you'll use a variable. There are two ways to do this. Be careful here because most programmers automatically go for the second approach, which is often not the right way.

Approach number 1: constant values

Sometimes the value to filter in the view does not change during the life of the view. For example, say you are coding a function where a number of records are being totaled based on an invoice number. For this example assume the invoice number is passed in a parameter called p_Invoice.

```
Thisview{PROP:Filter} = 'Items:InvoiceNumber = ' & p_Invoice
```

Notice the way the variable is outside the quotes. The PROP:Filter is a string, and you're assigning the value of p_Invoice into the string, rather than the variable name.

As long as p_Invoice is fixed for the life of the view this is more efficient because p_Invoice does not need to be evaluated for every cycle in the loop.

If p_Invoice is a string then you need to remember to use the single quotes around the string. I also usually CLIP the string to avoid problems with trailing spaces. For example;

```
ThisView{PROP:Filter} = |
  'Items:InvoiceNumber = <39>' & clip(p_Invoice) &'<39>'
```

Approach number 2: variable value

The second way to use a variable in the filter is to BIND it, and then use the name of the variable in the filter. Typically you'd do this if the value in the variable can change during the duration of the loop. For example;

```
BIND('Loc:Paid',Loc:Paid)
ThisView{PROP:Filter} = 'Inv:Paid = Loc:Paid'
```

Notice that in this example, the view engine is told how to handle the new variable (Loc:Paid) via the BIND statement. The new variable is then passed to the engine and it is evaluated for every record in the table, as the view is read. This code is slightly slower than approach number 1.

Server-side vs. client-side filtering

In order to get maximum performance from the view, in all cases, you need to differentiate between server-side filtering and client-side filtering. For ISAM you won't really notice a difference, but for SQL it makes an enormous difference. So get into the correct habit now.

The most common reason for a really slow browse when first porting to SQL is because client-side filters are included into the templates' Filter setting. If you have a browse that takes a long time to open, then check the filter very carefully.

Remember that the range and the filter are merged together into a single PROP:Filter statement. Thus if your filter contains *any* client-side component, it makes the whole PROP:Filter *including* the range client-side. For large tables this can mean the difference between fetching a dozen records or a million records.

Server-side filtering is anything that the server can do. Client side filtering is code that only your Clarion program can do. For example, SQL doesn't do arrays. So the filter:

```
'cus:lastname = <39>JOHNSON<39> and cus:FirstName[1] = <39>B<39>'
```

contains both a server side component and a client side component. However, because the server cannot do the whole filter it ends up doing none of the filter. In this case *all* the customers are included in the result set, and the file driver will apply the filter. This is grossly inefficient, both in terms of CPU resources and network resources.

If your view is going really slowly, then the first thing to do is remove the filter completely. If it then goes much faster you can add the filter back, one part at a time, until you discover which part is causing the problem. If that's a client-side bit then move that code into the LOOP structure.

In the example above the restriction on lastname can be done on the server, so set the PROP:Filter to include this restriction:

```
ThisView{PROP:Filter} = 'cus:lastname = <39>JOHNSON<39>'
```

Then inside the LOOP add a line to do the client side filter;

```
If cus:FirstName[1] <> 'B' then cycle.
```

What happens when you make a mistake?

I mentioned earlier that getting the filter right is the hardest part of using the view correctly. If you get the filter wrong then one of three things can occur;

1. The view returns either all records or no records. This is the most obvious effect, and the easiest one to spot. It is usually a simple error in the syntax of the filter itself. If the syntax is invalid then all the records in the view are returned. There may also be a logical error in the filter, in other words you may simply have not filtered out what you intended to filter out.

2. The view takes a really, really, long time to return the result set. But it does return the correct result set. This one is harder to spot because it becomes an issue of confidence. If you have confidence in the backend to return result sets quickly, and confidence in the view engine to do its job properly and efficiently, then you are more likely to notice when a view is running slowly.

3. The code "hangs" on the first NEXT statement in the loop. This usually happens when you've mismatched the field *type*. For example, if you forget the quotes around a string field, like this;

```
ThisView{PROP:Filter} = 'LastName = Johnson'
```

Advanced SQL-specific properties

There are a several other SQL properties with which you should be familiar.

PROP:SQL

PROP:SQL is a file property, not a view property. So if you are converting code beware of this and convert to one of the view properties if necessary.

PROP:SQLFilter

This property lets you add SQL-specific code to your filter. In other words if you have SQL code, not Clarion code, that you want added to the WHERE cause of the generated SELECT then you can add it using this property. Some care needs to be taken, however, and I recommend consulting the Clarion online help before using PROP:SQLFilter.

Obviously this code does not work with any non-SQL driver, so it does make your code less portable. However adding the SQL filter here, and then re-filtering inside the loop (using Clarion code), should give you the best of both worlds.

As with PROP:Filter though you may need to construct this additional filter very carefully. If you make a mistake, then the entire filter may fail.

Lastly, it's not just ISAM/SQL portability that's in jeopardy. Using PROP:SQLFilter may also affect the portability of your code from one SQL backend to another.

There is also a SQL() function in the Clarion language which performs the same functionality as PROP:SQLFilter, and which may be a better construction in some places. Read the Clarion documentation for more information on this command.

PROP:SQLOrder

PROP:SQLOrder behaves in the same way as PROP:SQLFilter. It allows you to include SQL specific code into the Order property of the view. This property is somewhat less useful though than the PROP:SQLFilter. In addition, because the view engine cannot decipher the code inside the PROP:SQLOrder, performance may be hindered rather than helped.

PROP:HINT

The MSSQL Server Driver and the Oracle Driver also support a rather cryptic property called PROP:Hint. This property is related to the HINT section of a SQL query, and is typically used to specify to the server which key to use, rather than allowing the server to select which key it thinks might be best. See your SQL server's documentation on hints for more ideas.

One place PROP:Hint is useful is to use it to limit the number of records which the SQL engine will return. For example;

```
    Invoices{PROP:Hint} = '&Top 5'
```

This will limit the server to returning the first five records. If you know how many records you will be needing (for example, if you are only interested in the first record in a result set) then alerting the SQL server to this limit can make a major impact on performance.

Note that the code assigned to PROP:Hint is SQL code native to the backend you are using, and valid code can vary from one database to another. For example, some databases use LIMIT not TOP to limit the number of records in the result set. Also note that, as of writing this, to my knowledge, the PROP:Hint property is *only* supported in the MSSQL and ORACLE drivers. Check the Clarion Driver documentation though as this property may be added to other drivers in the future.

Summary

A programmer recently told me that he tried views but discarded them because they didn't work or were orders of magnitude slower than looping through files. Another programmer told me he uses files because that's all he knows how to do.

Part of becoming better programmers is learning ways to improve the way we write code. Views do not offer new functionality, rather they offer a better way of implementing functionality that we already use on a daily basis. Views, done badly, hinder rather than help, and indeed performance is very poor. Views done *correctly* will make your program more efficient, more flexible and more portable.

There is far more that you can learn about views and VIEW structures, so remember to read the Clarion documentation on the subject.

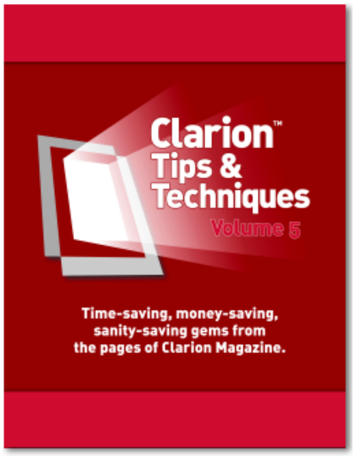
Living in Cape Town, South Africa, Bruce Johnson is a part-owner of CapeSoft and has been programming in Clarion since 1992. He authored the successful "Programming in Clarion's ABC" book and has been involved in some of Clarion's most popular accessories. When not programming he enjoys cooking, and sports - the one as a direct result of the other.

Reader Comments

Add a comment

Clarion Magazine

Clarion Tips & Techniques Volume 5



Clarion Tips & Techniques Volume 5 is another 650+ page blockbuster brimming with terrific articles from Clarion Magazine.
ISBN: 978-0-9784034-1-6

Buy now in the Clarion Magazine store!

Table of contents

- **The Clarion Language**

- **User Interface**

- **WinAPI**

- **Windows Vista**

- **Internet**

- **Templates**

Purchase now in the Clarion Magazine store!

Source code

ClarionTipsTechniquesVolume5Source.zip (3 MB)

Errata

Corrections to the book will be posted here as necessary. If you have found an error please report it.

Clarion Magazine

Migrating From MySQL To PostgreSQL, Part 1

by Dave Harms

Published 2009-02-20

For years, I've used MySQL as the data store for Clarion Magazine. And really I have little to complain about; MySQL is fast, it is increasingly capable (although I'm still on the relatively old 4.1 release), and it's been quite reliable.

But I'm switching to PostgreSQL.

In this article and the next I'll describe the process by which I'm migrating the ClarionMag database, and I'll cover some of the conversion issues I've discovered along the way.

First, however, I need to answer one question: Why switch?

Why PostgreSQL?

If I'm happy with MySQL, why would I change to PostgreSQL? The short answer is that I feel I'm ready to graduate to a more robust SQL database, one with a long pedigree of core SQL functionality including transactions, triggers, stored procedures, subqueries and the like.

Now, for you MySQL fans howling in protest, let me say that yes, MySQL has in recent years achieved much of PostgreSQL's functionality. But to me there's an important caveat to MySQL's feature list. One of the most important aspects of any serious SQL database is transaction support, which guarantees data integrity. MySQL own transactional engine is still in beta; to get the real deal you have to go to a third party product.

Transactions and MySQL

While PostgreSQL is, like most SQL databases, a combination of a query language and a storage engine, MySQL has an architecture that separates the storage engine from the query language. That separation can be an advantage, since it means vendors can create special-purpose storage engines. MySQL's default storage engine is MyISAM, which stores data in ISAM files. But MyISAM doesn't support transactions. Storage engines that do support transactions include BDB (obsolete), SolidDB, PBXT, InnoDB and Falcon.

By far the most widely used transactional storage engine for MySQL, and the one typically referenced in benchmark testing, is Innobase's InnoDB.

InnoDB is an impressive piece of software; so impressive, in fact, that Oracle bought Innobase (and InnoDB) in 2005. (Oracle had previously attempted to buy MySQL but was rebuffed.) The purchase of InnoDB by Oracle caused not a little consternation in the MySQL community, since MySQL+InnoDB was seen by some as a competitor to Oracle.

For now, at least, InnoDB is still available under the same terms as before, but as it is owned by Oracle there's no guarantee as to future costs or availability. MySQL is clearly concerned; they've been working hard on Falcon, their in-house replacement for InnoDB.

Falcon is coming along, but it's still a beta product. I suppose I could go with InnoDB now and then migrate to Falcon at some future point if Oracle gets stroppy about licensing, but on the whole I'd prefer to choose something now that I can expect to use in five to ten years time.

Comparisons

It's not hard to find web pages purporting to compare MySQL and PostgreSQL; unfortunately most of the comparisons are outdated. Be careful and always check the publication date. Older comparisons will tell you that PostgreSQL is slow, has no native Windows version, and has many SQL capabilities lacking in MySQL, while MySQL is fast, unreliable, and feature-poor. The reality is that these products are converging in many ways. PostgreSQL's peformance has improved significantly, and even the Windows version looks pretty solid; MySQL has most of the features you'd expect in a full-fledged SQL database (with InnoDB, that is).

One of the best comparisons I've seen is at WikiVS.com. You may also want to have a look at the "gotchas" lists for both MySQL and PostgreSQL.

My key issues in the ongoing MySQL vs PostgreSQL battle:

- PostgreSQL benefits more than MySQL from tuning; the default install is geared toward minimal available memory. But tuning can be tricky.
- Speeds are comparable, InnoDB to PostgreSQL. But PostgreSQL scales better when you start throwing more processors at it. PostgreSQL also scales better when queries are more complex.
- PostgreSQL has integrated full text search (although I may end up sticking with Lucene for text search capabilities)
- PostgreSQL excels at maintaining data integrity.
- PostgreSQL is completely free

On the other side, I'm somewhat concerned about replication in PostgreSQL. I keep a backup server online at all times, but I only have a manual process for replicating the data that involves taking a nightly dump of the MySQL database. The irony is that MySQL supposedly has decent replication, and I've never tried it.

The most commonly-used replication solution for PostgreSQL is Slony-I, which is a bit resource-hungry as it relies on triggers. Bucardo looks promising, but isn't available for Windows (although if I had very high volumes of data I'd probably be considering a Linux-based PostgreSQL installation anyway). Realistically, all I need at present is a warm standby solution.

There are other points to be made on both sides, as you'll discover as you read the links. But these are the issues that matter most to me.

Future development

MySQL has always been a company-owned product, with dual open source and commercial licensing; PostgreSQL, on the other hand, is a solely open source product that originated at the University of California, Berkley. PostgreSQL is managed by a distributed but very stable core team. Many of these folks are paid by their companies to work on PostgreSQL. According to Josh Berkus, "80% to 90% of the code changes that go into any given version are written by people who were either paid directly to work on PostgreSQL development, or for whom working on PostgreSQL development is an approved use of their work time."

Sun Microsystems (where Berkus formerly worked) is often seen as one of the driving forces behind PostgreSQL development. In 2008 Sun paid a princely one *billion* dollars for MySQL AB, which seemed to call into question the company's commitment to PostgreSQL. Sun's people insist the company is happy with both products and is still backing PostgreSQL development.

There have been a couple of high-profile departures from Sun's MySQL team, notably MySQL founder Monty Widenius. Widenius has been critical of Sun releasing MySQL 5.1 before it reached stability:

> The reason I am asking you to be very cautious about MySQL 5.1 is that there are still many known and unknown fatal bugs in the new features that are still not addressed.

Widenius reportedly Sun left on good terms and has founded a new company, Monty Program AB.

Licensing

There's a common misconception that MySQL, being open source, is also free. This is not necessarily the case. MySQL is available under either a commercial license or under the Gnu Public License (GPL). The GPL states that any software that makes use of GPL software automatically comes under the GPL, and if you distribute any GPL'd software you also have to distribute the source code. If, say, you create a point of sale program that uses MySQL, and you want to include or use the free version of MySQL, you have to also provide your customers with the source code for your application. If you don't want to do that you need to get a commercial license for MySQL.

The key to understanding MySQL's licensing is the word "distribute." If you distribute software that uses MySQL, whether to customers on the other side of the world or simply to someone else in your own company, you must either make your application's source code available or you must purchase a commercial license.

One oddity of this arrangement is you don't need a commercial license to use MySQL as the data store for a web site, since you're not actually distributing your application.

Or perhaps it's not so simple after all. What if your web site makes use of client-side Javascript? At that point you're distributing an application to a user, and it could be that you're obligated to distribute all of the web server source code as well. As far as I know this issue has not been clearly resolved, and the standard GPL license makes no special exception for Javascript.

In contrast, PostgreSQL is released under the permissive BSD license. You can use and distribute PostgreSQL as you like, provided you maintain the copyright notices and do not use PostgreSQL team members' names in a promotional way without permission. You can even create your own commercial version of PostgreSQL, as EnterpriseDB and other companies have done.

Converting the database

In an ideal world, all you'd need to do to convert from one SQL database to another is dump the source database as SQL statements. Here's a dump of a very simple table I created for an article series on embed points:

```
DROP TABLE IF EXISTS `embedchain`;
CREATE TABLE `embedchain` (
  `EmbedChainID` int(11) NOT NULL auto_increment,
  `Chain` varchar(20) default NULL,
  PRIMARY KEY  (`EmbedChainID`),
  UNIQUE KEY `EmbedChainID` (`EmbedChainID`)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

LOCK TABLES `embedchain` WRITE;
INSERT INTO `embedchain` VALUES (1,'Clarion'),(2,'ABC');
UNLOCK TABLES;
```

This table only has two columns, the equivalent of a LONG and a CSTRING, so it should be a snap to convert, right?

If I attempt to execute this SQL verbatim in PostgreSQL I get the following error:

```
ERROR:  syntax error at or near "`"
LINE 1: DROP TABLE IF EXISTS `embedchain`;
```

It appears PostgreSQL doesn't like backward single quotes. I really don't need them, so I'll simply do a search and replace and get rid of them. You probably don't want to just change those backward single quotes to regular single quotes either. As I noted in PostgreSQL Revisited:Porting From TPS:

> PostgreSQL automatically converts any labels in any SQL statements to lower case. This results in a kind of case insensitivity. If you have a field called lastname, you can write SQL that refers to it as lastname, Lastname, LastName, or lAStnaME for all I care, and PostgreSQL will convert your label to lastname, and all will be well. But if you use any upper case characters in your labels you'll need to specify those labels in quotes or PostgreSQL will give you an error, even if what you type exactly matches the case of the label. And that can be a real pain, because you'll need NAME attributes on everything and you'll have to add quotes around labels whenever you write embedded SQL.

My SQL now looks like this:

```
CREATE TABLE embedchain (
  EmbedChainID int(11) NOT NULL auto_increment,
  Chain varchar(20) default NULL,
  PRIMARY KEY  (EmbedChainID),
  UNIQUE KEY EmbedChainID (EmbedChainID)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
```

Since I'm not using quotes I can refer to my table and columns in upper case, lower case or mixed case and PostgreSQL will internally convert everything to lower case.

```
ERROR:  syntax error at or near "("
LINE 2:   EmbedChainID int(11) NOT NULL auto_increment,
```

It turns out that int(11) isn't a valid data type in PostgreSQL; the equivalent is simply integer. With that changed the error becomes:

```
ERROR:  syntax error at or near "auto_increment"
LINE 2:   EmbedChainID integer NOT NULL auto_increment,
```

MySQL and PostgreSQL have different autonumbering concepts. In MySQL you simply use the auto_increment attribute to make a column autonumbering. In PostgreSQL you create something called a *sequence*. But let's forget about that for a moment and just try to create the table. Here's the statement so far:

```
CREATE TABLE embedchain (
  EmbedChainID integer NOT NULL,
  Chain varchar(20) default NULL,
  PRIMARY KEY  (EmbedChainID),
  UNIQUE KEY EmbedChainID (EmbedChainID)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
```

Does it work yet? Not a chance:

```
ERROR:  syntax error at or near "KEY"
LINE 8:   UNIQUE KEY EmbedChainID (EmbedChainID)
```

How about like this:

```
CREATE TABLE embedchain (
  EmbedChainID integer NOT NULL,
  Chain varchar(20) default NULL
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
```

Of course not. But this works:

```
CREATE TABLE embedchain (
  EmbedChainID integer NOT NULL,
  Chain varchar(20) default NULL
) ;
```

Of course there's no autoincrementing, there are no keys, etc. But on to the data:

```
LOCK TABLES embedchain WRITE;
INSERT INTO embedchain VALUES (1,'Clarion'),(2,'ABC');
UNLOCK TABLES;
```

No, PostgreSQL doesn't care for the LOCK TABLES, although it does have a LOCK TABLE statement, but that has to be used inside a BEGIN / COMMIT block. Okay, let's forget about locking:

```
INSERT INTO embedchain VALUES (1,'Clarion'),(2,'ABC');
```

Hey! That actually worked!

But that's a lot of work for a teeny tiny two-column table, and the embedchain table is not, in fact, fully specified.

Here's what the embedchain create statement *should* look like:

```
CREATE SEQUENCE embedchain_embedchainid_seq  START WITH 3 ;


CREATE TABLE  embedchain (
   embedchainid integer DEFAULT nextval('embedchain_embedchainid_seq') NOT NULL,
   chain   varchar(20) default NULL,
   primary key (embedchainid),
 unique (embedchainid)
)   ;
```

This SQL code creates a sequence, which supplies autonumbering via the nextval statement. The CREATE TABLE statement will also trigger the creation of a unique index on EmbedChainID.

What do you think? Easy, isn't it?

The point here is not that it's a pain porting *to* PostgreSQL; it's that it's almost always a pain porting *between* any two flavors of SQL. As Mike Gorman likes to say, there is no such thing as standard SQL. Every vendor does things differently.

As the above example illustrates, there are a minimum of two tasks involved in converting from one SQL database to another. The first is to convert the schema, which includes table definitions, indexes, triggers, etc.; The second is to convert the data.

Converting the schema

Getting a dump of the MySQL schema is easy with the mysqldump utility. I used this command to write the schema to a file:

```
"C:\Program Files\MySQL\MySQL Server 4.1\bin\mysqldump" ↵
  -d -u root -p cmag > mysqlschema.sql
```

By default mysqldump will give you the schema and the data. The -d option tells mysqldump not to dump the data.

Next I needed to convert the schema to PostgreSQL's format.

It's far better not to have to reinvent the wheel. I found a Perl script called *mysql 2 pgsql* over at PGFoundry. I don't normally work with Perl and didn't have an interpreter handy, so I also downloaded and installed Strawberry Perl.

I ran the MySQL schema through mysql2pgsql.perl and had a look at the output. Immediately I noticed that the conversion script had put all table and field names in double quotes, which would force case sensitive matching on names as noted above. I didn't want that so I did a search and replace and got rid of all the double quotes.

I tried running the create script against my PostgreSQL database, at which time I discovered a few additional incompatibilities.

MySQL lets you create table names beginning with a number, but that's against the rules in PostgreSQL. Rather than deal with this in the conversion I decided to fix the problem at the source, in the MySQL database. Since the table in question wasn't being actively used I simply issued the following:

```
rename table 3PProduct to TPProduct;
```

The only column name that caused problems was user which is a reserved word in PostgreSQL. I decide username would be a good substitute. And while I needed to make this change on the live site, I didn't want to bring the site down for any lengthy maintenance.

The easiest solution was to create a new column called UserName immediately after the User column:

```
alter table <tablename> add column UserName varchar(50) after user;
```

(This is one feature I quite like in MySQL; if there's a way to specify the location of a column in PostgreSQL I haven't found it. Newly created PostgreSQL columns are always appended to the end of the table.)

I copied the contents of the User column to the UserName field, and when I was satisfied that all my code was correctly referencing the UserName I used the ALTER TABLE command to remove the User column.

```
alter table <tablename> drop column user;
```

In other circumstances I might have to compile the code, take the web server offline, execute a script to update the database, and then restart the web server with the new code. And in a situation like that I'd probably want to test the process on a test database first.

With the schema conversion well in hand I was ready to tackle actually converting the data. I'll go through that process next time.

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

Add a comment

Clarion Magazine

Migrating From MySQL To PostgreSQL, Part 2

by Dave Harms

Published 2009-02-20

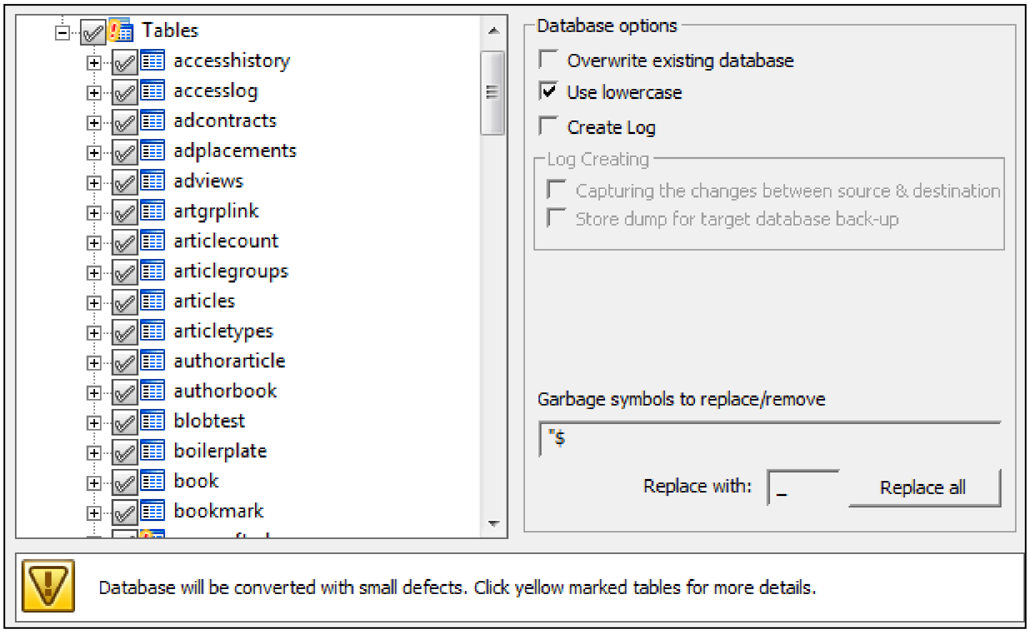
In Part 1 I gave my rationale for migrating from MySQL to PostgreSQL and illustrated the kinds of problems you're likely to face any time you migrate from one SQL database vendor to another. Now it's time to actually start moving the data.

Moving the data

I considered writing my own code to port the data from MySQL to PostgreSQL. I looked into the C6 conversion program wizard, but I couldn't get it to work (perhaps someone who's had better luck will consider writing an article...). I thought about writing a template to generate my own conversion program. It wouldn't have been all that difficult, but in the end it came down to a question of why build when I could buy?

There are a number of tools out there that will copy data from one database to another. The one I settled on is DBSync for MySQL & PostgreSQL from the folks at DBConvert. This is a commercial product, but quite reasonably priced at $79. DBConvert also makes a conversion tool, and had I found the company before I'd achieved the schema migration I might well have purchased their MySQL - PostgreSQL conversion tool instead. In fact, DBSync will also create the destination database if you wish, although it doesn't do as complete a job as mysql2postgresql.perl (specifically, it didn't create any timestamp triggers).

In Figure 1 I've already defined my database connections and I've selected all tables for synchronization. Note that I've checked the Use lowercase option. If you don't select this option DBSync will create all object names in quotes, forcing case sensitivity.



☑ Tables
- ☑ accesshistory
- ☑ accesslog
- ☑ adcontracts
- ☑ adplacements
- ☑ adviews
- ☑ artgrplink
- ☑ articlecount
- ☑ articlegroups
- ☑ articles
- ☑ articletypes
- ☑ authorarticle
- ☑ authorbook
- ☑ blobtest
- ☑ boilerplate
- ☑ book
- ☑ bookmark

Database options

☐ Overwrite existing database

☑ Use lowercase

☐ Create Log

Log Creating

☐ Capturing the changes between source & destination

☐ Store dump for target database back-up

Garbage symbols to replace/remove: "$

Replace with: _

Database will be converted with small defects. Click yellow marked tables for more details.

Figure 1. DBSync database options

DBSync is fast, as it uses native APIs rather than ODBC. And it quickly let me know that there were a number of problems with the MySQL data when it came to enums, which are analogous to Clarion's "must be in list" validation requirement.

Here's a MySQL enum declaration (excerpted from a CREATE TABLE statement):

```
`RequireRights` enum('Y','N','A','P','E','Z')
  default 'Y',
```

MySQL's enums are like CHECK constraints, where a column can have only one of a specified set of values. These are a bit like equates; internally MySQL stores the index of the selected value, not the value itself.

My enums got converted to PostgreSQL CHECK constraints:

```
requirerights varchar CHECK (requirerights IN
  ( 'Y','N','A','P','E','Z' )) default 'Y',
```

Now, up until MySQL 4.1 (the version used by ClarionMag) enums weren't really constraints at all; it was possible to enter bad data. But PostgreSQL wasn't having any of that, and I ended up writing about twenty SQL UPDATE statements to run against the MySQL database before I did the import. Here's a typical example:

```
update Articles set status = 'A' where status = '';
```

Cleaning up the new database

There are two applications that use the ClarionMag database: the web application, written in Java, and an admin application, written in Clarion (both of which I'm in the process of replacing with a Clarion# web application). When I switched the Java web application over to the PostgreSQL database I discovered some new issues.

The existing Java web app (I'm sorry to say) contains a fair bit of hard coded SQL. Quite a bit of that code worked fine, but some didn't. For the most part there were two kinds of problems: missing functions and incorrect data types.

Missing SQL functions

My SQL code made use of a number of MySQL-specific functions, such as date_format, to_days, and concat. Those functions don't exist in PostgreSQL, so I have two options: recreate the functions or rewrite the code.

Over at PostgreSQL.org you can find a set of MySQL compatibility tools. From the README:

> This project is a collection of functions, aggregates, operators and casts that make PostgreSQL mimic MySQL as closely as possible.
>
> To use the project, you can either find and install the few functions that you need, or run all the .sql files to install the complete compatibility environment.

There are a few things the compatibility library can't handle, including: returning 0 or 1 instead of t or f for boolean operators (although PostgreSQL does have a boolean cast); interval type differences; and case-sensitive string comparisons.

PostgreSQL strings are always case sensitive, whereas MySQL strings are case insensitive by default. I found one place in my code where I issued a SELECT using a mixed-case string in an equality test. The actual data was upper case; in MySQL the SELECT returned the desired value; in PostgreSQL no record was retrieved. Changing the SELECT statement to use the matching uppercase value fixed that problem.

Data types

A long time ago I had problems with MySQL decimal columns and Clarion, and as an interim solution I began using CSTRINGs (VARCHARs) instead. My interim solution, as often happens, became permanent.

My Java code never minded converting MySQL VARCHARs to double precision integers on the fly, but once I switched to the PostgreSQL JDBC driver I began getting Java errors indicating that my CStrings were a "bad value for type double". It was time to switch back to using decimals.

For instance, there's a VARCHAR(9) in my original Coupon table called Value. When I attempted to convert that field to a decimal using this statement:

```
alter table coupon alter column value type decimal(10,2);
```

I got an error:

```
column "value" cannot be cast to type "pg_catalog.numeric"
```

To get around the error I had to do an explicit cast:

```
alter table coupon alter column value type decimal(10,2)
  using cast (value as decimal(10,2));
```

That worked for quite a few of the columns I needed to convert, but not all of them. PostgreSQL's pretty careful with your data - when you attempt to change the data type of a column, using an explicit cast, it checks to make sure all the data can actually be converted to the new time. If it can't convert some data it aborts the type change. If the string to be cast was a valid decimal or integer everything was cool. But in some cases the data was bad; if at least one row had an empty string in that column I was unable to change the column's type.

I could have written a quick SQL statement to change the empty strings to '0' in each case, but instead I set up a single stored procedure to handle the cast and accommodate the special case of an empty string:

```
CREATE OR REPLACE FUNCTION CastVarcharToDecimal(oldValue character varying)
  RETURNS decimal(10,2) AS
$BODY$
SELECT
    CASE WHEN $1=''
    THEN 0
    ELSE cast($1 as decimal(10,2))
    END
$BODY$
  LANGUAGE 'sql' IMMUTABLE STRICT;
```

After running this SQL to create the stored procedure I changed my type conversion statement as follows:

```
alter table coupon alter column value type decimal(10,2)
  using CastVarcharToDecimal(value);
```

But then I got another error:

```
default for column "handling" cannot be cast to
  type "pg_catalog.numeric"
```

Hmm. I thought I'd fixed that with the cast! But then I took a closer look at the error. It wasn't telling me I had a problem with the existing data. Rather, I had a problem with the default data. Here's the original declaration:

```
handling   varchar(5) default NULL,
```

I changed the definition to drop the default:

```
alter table shipping alter column handling drop default;
```

and I was able to convert the column type.

Fixing the dictionary

Along with converting the dictionary and getting the Java web app working, I had to do some massaging of my data dictionary. As noted earlier, PostgreSQL defaults all table and column labels to lower case. But my dictionary contained a great many NAME attributes.

All I had to do to fix up column names was delete the NAME attribute. But I wasn't about to do that manually for each and every column. Instead, I exported a TXD and used Clarion 7's regex search and replace feature. I used this as a search term:

```
\,NAME\(\'.*\'\)[^,PRE]
```

I left the replacement field blank, so search and replace just removed the NAME attribute. I created a new dictionary from the TXD, switched the app to use that dictionary, and I was off to the races. (I also had to set up a suitable ODBC data source, but in the MySQL version I used a global variable for the owner name, so that wasn't a big problem.)

Table names

After creating the new dictionary from the TXD I fixed up all the table names manually. It's an oddity of the file driver system that if you don't specify a Full Pathname (which gets generated as a NAME attribute) for a file/table then only the first eight characters will be used. For instance, a table called AuthorArticle will be abbreviated to AuthorAr, which PostgreSQL will see as authorar (unless your table name is in quotes). The proper value in this case is authorarticle, entered in the Full Pathname field in the Edit Table Properties window. (My tables are all in the public schema, so while I could also call this table public.authorarticle the public schema is assumed.)

Summary

When I started out writing the ClarionMag web app I didn't anticipate changing SQL servers. I was just happy to take advantage of the power of SQL any way I could, and that largely took the form of hard-coded SQL statements. Converting the ClarionMag database from MySQL to PostgreSQL exposed the weaknesses in this approach. As Bruce Johnson recently pointed out in his article on views, hard coded SQL statements (whether via PROP:SQL or, in my case, Java code using JDBC) tend to limit portability. And you never know when circumstances will force a change in database vendors.

The moral of the story: Write for portability! In Clarion, use views as much as possible. Clarion# has views as well, but opens up some other possibilities such as using an object-relational mapping (ORM) layer like NHibernate (but that's well beyond the scope of this article).

And if like me you need to migrate from one SQL vendor to another, do some research first. Most likely a whole lot of people have traveled that path before you, and you'll be able to find some reasonably-priced (or free) utilities to make the migration as painless as possible.

Resources

- WikiVS: MySQL vs PostgreSQL
- WikiBooks: Converting to PostgreSQL
- Changing column data types
- The MySQL compatibility library
- DBConvert/DBSynch

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

Add a comment

Clarion Magazine

C7 AppGen Build 4869 (updated)

by Dave Harms

Published 2009-02-25

Updated Feb 25, 2009: Ben Dell pointed out to me that my comments on TXDs suggested that you can use them to backport a dictionary to C6.3. That may or may not be true.

C7 beta build 4869 went out to CSP participants on Feb 13, and contained by far the longest list of bug fixes and new features of any release to date. A number of these were cosmetic, and it's always nice to see a beta progress to the stage where the team starts hitting on the lower priority issues. But there were lots of meaty fixes as well.

Areas affected include:

- Data diagrammer
- Macros
- Crashes and errors while browsing tables
- Redirection
- Linker
- Listbox formatter

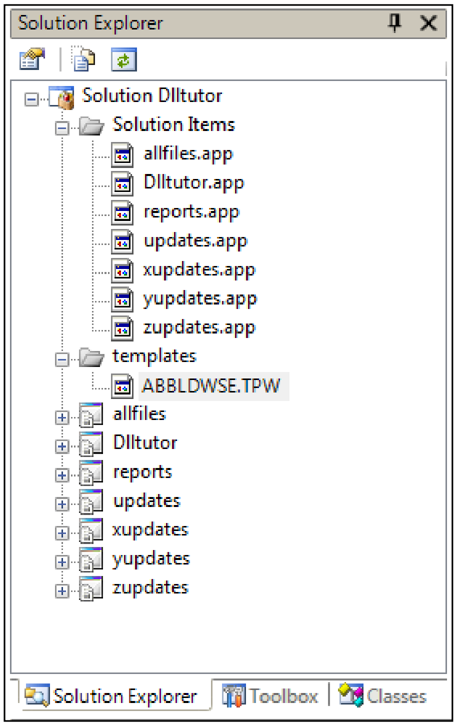
Now for a closer look at some of the other fixes and features...

Solution folders and template editing

Somewhere along the line, perhaps in a previous build, C7 gained the ability to add solution folders. Clarion# has always let you add new folders to a solution, and those are actual folders in the file system. When you add a file to a solution's folder and the file isn't already in the actual directory, Clarion# asks you if you want to copy the file or create a link.

C7's folders are virtual folders; any file you add to a folder you create is linked, not copied.

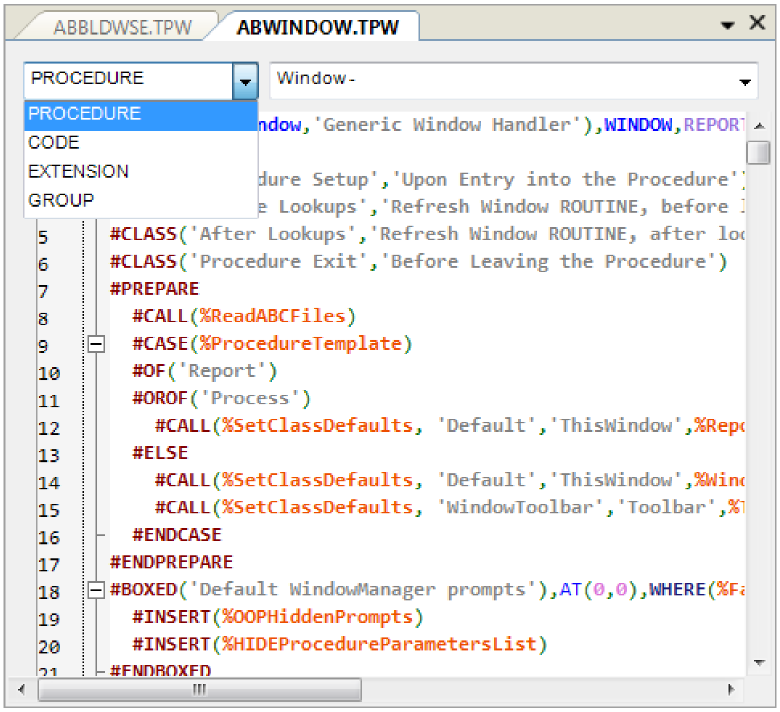
One use for these virtual folders is as a way to keep templates handy, assuming you're someone who codes templates. In Figure 1 I've created a solution folder called templates, and I've added one of the C6 ABC template files. I can now double-click on the template file to edit it.



```
Solution Dlltutor
  Solution Items
    allfiles.app
    Dlltutor.app
    reports.app
    updates.app
    xupdates.app
    yupdates.app
    zupdates.app
  templates
    ABBLDWSE.TPW
  allfiles
  Dlltutor
  reports
  updates
  xupdates
  yupdates
  zupdates
```

Figure 1. A solution folder with a template file.

Speaking of templates, C7 now has specialized support for editing templates. Figure 2 shows abwindow.tpw opened in the editor. The left drop list shows the types of template sections available in the file.

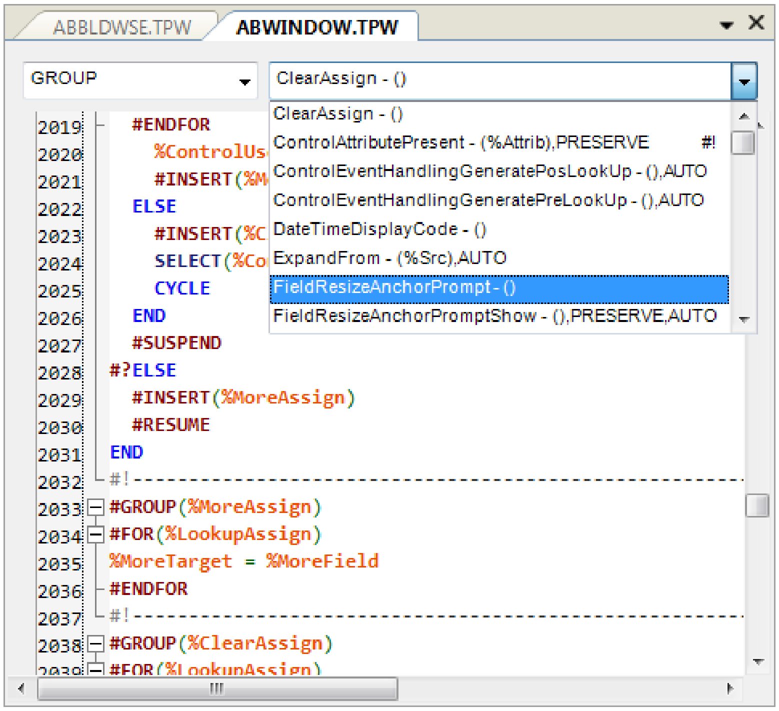


```
ndow,'Generic Window Handler'),WINDOW,REPORT

dure Setup','Upon Entry into the Procedure')
e Lookups','Refresh Window ROUTINE, before l
#CLASS('After Lookups','Refresh Window ROUTINE, after lo
#CLASS('Procedure Exit','Before Leaving the Procedure')
#PREPARE
  #CALL(%ReadABCFiles)
  #CASE(%ProcedureTemplate)
  #OF('Report')
  #OROF('Process')
    #CALL(%SetClassDefaults, 'Default','ThisWindow',%Repo
  #ELSE
    #CALL(%SetClassDefaults, 'Default','ThisWindow',%Winc
    #CALL(%SetClassDefaults, 'WindowToolbar','Toolbar',%T
  #ENDCASE
#ENDPREPARE
#BOXED('Default WindowManager prompts'),AT(0,0),WHERE(%Fa
  #INSERT(%OOPHiddenPrompts)
  #INSERT(%HIDEProcedureParametersList)
#ENDBOXED
```

Figure 2. List of template types

Select a template type and the right drop list shows the available templates (Figure 3).


ABBLDWSE.TPW
ABWINDOW.TPW
GROUP
ClearAssign - ()
ClearAssign - ()
ControlAttributePresent - (%Attrib),PRESERVE #!
ControlEventHandlingGeneratePosLookUp - (),AUTO
ControlEventHandlingGeneratePreLookUp - (),AUTO
DateTimeDisplayCode - ()
ExpandFrom - (%Src),AUTO
FieldResizeAnchorPrompt - ()
FieldResizeAnchorPromptShow - (),PRESERVE,AUTO
III

```
  #ENDFOR
    %ControlUs
    #INSERT(%M
  ELSE
    #INSERT(%C
    SELECT(%Co
    CYCLE
  END
  #SUSPEND
#?ELSE
  #INSERT(%MoreAssign)
  #RESUME
END
#!----------------------------------------------------
#GROUP(%MoreAssign)
#FOR(%LookupAssign)
%MoreTarget = %MoreField
#ENDFOR
#!----------------------------------------------------
#GROUP(%ClearAssign)
#FOR(%LookupAssign)
```

Figure 3. Locating a #GROUP

Also note that there is support for code folding in templates as well. Add in bookmarks and you have an impressive toolset for template writers.

Other template changes include a global option to set the TRN attribute on every OPTION,GROUP,RADIO,STRING, CHECK,PROMPT controls that are inside a TAB control.

As well, there are now several quick starts to create a new Clarion template, but the ABC and Clarion versions are a bit buggy. As in, they don't work.

There are a couple of new embed points, one for legacy compatibility in the data section, and another called %WindowStructureForEachControls to preprocess every control.

Memory usage

A few folks have reported high memory usage in this build. According to Bob Z, this isn't a regression. Rather the problem is that apps are cached for quick reloading as long as the solution is open. So if you open a whole bunch of apps in a large solution, you're going to have memory issues. Z has offered a couple of solutions which revolve around optionally removing or managing the cache.

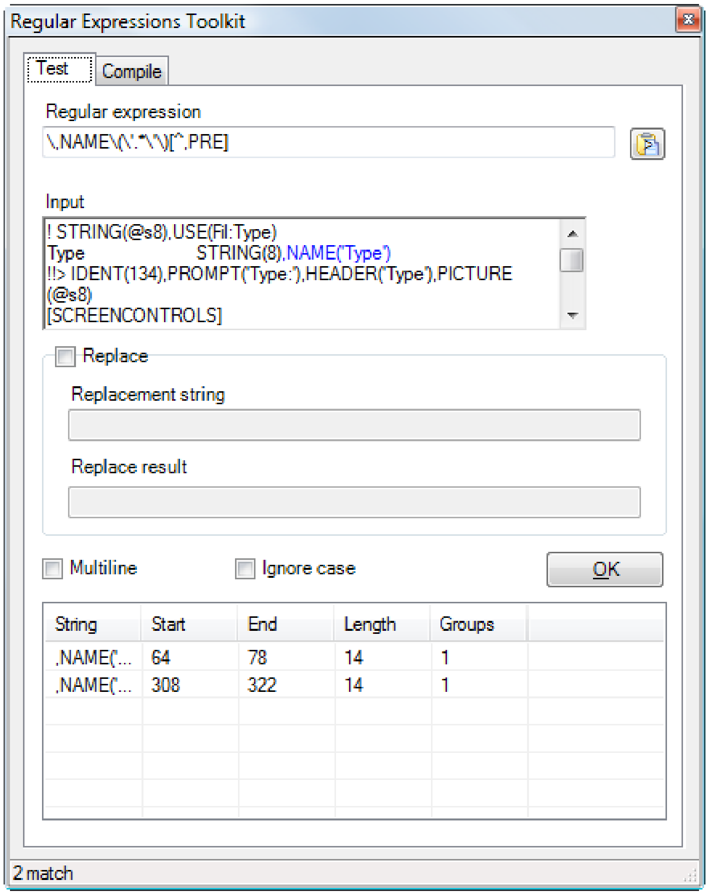
While a solution along those lines is welcome, at the same time I think it's worth remembering that you can attempt much larger, more complex tasks in C7 than you can with C6, thanks to multi-DLL solutions. Memory usage is inevitably going to be an issue. Ultimately, the best solution for truly massive multi-DLL applications may simply be to have a development machine suited to the task, meaning a 64 bit version of Windows and as much RAM as is necessary. But by all means, let's have interim solutions.

Misc stuff

Window field equates are now visible in code completion. The list appears as soon as you type a letter following the ?, or you can press Ctrl-Space to bring up the entire list of code completion options (as before).

You can now populate two browses on a window and have their update procedures called correctly. If you have an existing window with this problem you need to delete the second one's update button template, close designer, open designer and repopulate

One of the fixed PTSS reports mentioned the Regular Expressions Toolkit, which is one of the things C7 inherits from SharpDevelop. I hadn't actually noticed this before, or if I had I'd forgotten about it. Have a look at it - it's under the Tools menu. You can test out your regexes, and you can compile them to .NET assemblies. I wish I'd known about this a couple of weeks ago; I'd have saved myself some trial and error in a TXD fixup while porting from MySQL to PostgreSQL....



Regular expression

```
\,NAME\(\'.*\'\)[^,PRE]
```

Input

```
! STRING(@s8),USE(Fil:Type)
Type                    STRING(8),NAME('Type')
!!> IDENT(134),PROMPT('Type:'),HEADER('Type'),PICTURE
(@s8)
[SCREENCONTROLS]
```

□ Replace

Replacement string

Replace result

□ Multiline □ Ignore case

String	Start	End	Length	Groups
,NAME('...	64	78	14	1
,NAME('...	308	322	14	1

2 match

Figure 4. The Regular Expressions Toolkit

Old fashioned TXDs are back in town, but with an asterisk. Although previous releases could export apps as TXAs, the only way to export a dictionary was as an XML DCTX file. According to the release notes, the TXD export was restored but only for Report Writer compatibility (there isn't a C7 Report Writer yet; you have to use the C6 version). And the format of the TXD is slightly different. The very first line of the C7 RR TXD is

[VERSION2]

which isn't valid for Clarion 6.3. If you delete that line you can get the TXD to import into a DCT (at least I can in 9055), but in my tests there are some differences in how relations are declared. It's not clear to me if that's a bug in the export code or a legitimate difference. If you attempt to use the TXD to return dictionary changes to C6.3 I advise you to generate source code in C7, using the C6.3 templates, and then compare that source code with what you get in C6.3. Clear as mud?

At the moment, exporting a TXD is a little non-obvious. First, open a dictionary. Now, click on the Import/Export button (Figure 5) and choose Export Dictionary to text.

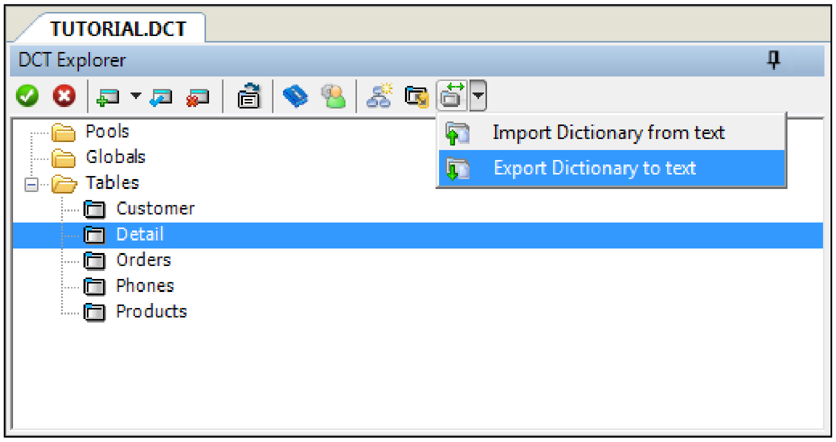


Figure 5. Exporting a TXD

Now, here's the tricky part. In the save dialog the default file type is set to Dictionary Text (*.dctx). Change that to All files (*.*) and type a file name ending in .txd. You must choose All files and you must use a .txd extension to get a TXD, otherwise you'll get an XML file.

Every since the first beta release of the AppGen I've been harping on the inability of the IDE to go to embeds on finding compile errors in embedded code. In this release that functionality is implemented, if incomplete.

If the compile error is in code that's in an embed point then double-clicking on the error in the Errors pad will open the embeditor to the correct embed. However as soon as you do this the Errors pad clears. If there's more than one error there's no way that I can see to get to the next error. PTSS 32344.

This is a big step in the right direction, however.

Things not yet fixed

There are still a number of open PTSS reports, as well as those filed since this build was released. As of this writing there are 173 confirmed, still open, non-private (visible to other beta testers) bug reports going back about two years to the very first beta release of C7 (yes, it has been two years). Just for fun I thought I'd go through the oldest of these (my comments are in italics).

1. Inconsistent cursors used when clicking and placing controls on a window. *Definitely a bug, but I'm more likely to drag/drop which works better anyway*.
2. Some child windows display in the middle of the monitor (as when you click on a button's Edit Text option. *I suppose this can be an issue if your IDE is spread over multiple monitors*.
3. A somewhat obscure problem involving views, CStrings, trailing spaces, PREVIOUS, NEXT, and ORDER. Affects C6 and C7 (*so not actually related to the new IDE*).
4. When using the keyboard, not the mouse, to select a drop-down property from a non-pinned, docked property list (it flies out from the side when you want it), the property list closes prematurely.

Obviously, as you go down this list the bugs are going to get more serious (yes, Lee, I know, keyboard issues *are* serious). But clearly not every outstanding bug is a showstopper.

The top five areas for non-private bug reports, with the highest number of reports first, are:

- IDE
- AppGen
- Structure designer
- RTL

- Dictionary Editor

Those five areas account for 75% of the reports. Areas with few issues are:

- Source editor
- Project system
- Data/ tables pad
- Template language
- File drivers
- Compiler/linker
- Control templates pad
- Extended UI
- Data diagram
- Browser
- Debugger

Steady as she goes

With numerous bug fixes and the welcome addition of error handling, this beta release is a big step in the right direction. I don't think C7 is quite at candidate release status yet (the beta Quick Start docs notwithstanding), but that day is at least in sight.

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, February 27, 2009 by Dave Harms

Ben Dell has had confirmation of an fix to handle the memory usage issues with multiple apps. In the next release there will be an option to disable automatic caching (lazy loading) of applications in a solution. This option will be available under Tools > Options > Clarion > Application General Options.

In SV's testing of Ben's solution memory usage stayed constant while each app was opened in succession.

Add a comment

Clarion Magazine

Building C7 Apps With MSBuild

by Dave Harms

Published 2009-02-28

The process by which Clarion 7 compiles and links your programs is significantly different from the process in C6 and earlier releases. Those versions of Clarion use SoftVelocity's own project system; C7, like Clarion#, uses Microsoft's MSBuild. In Understanding The C7 Build System I covered the basic concepts of projects and solutions, but I didn't say much about the MSBuild utility itself. After all, you don't really need to understand MSBuild to use C7 since the IDE handles all that for you.

On the other hand, calling MSBuild directly opens up a whole new world of opportunities.

What is MSBuild?

MSBuild is an executable program included with the .NET framework. Its main purpose is to manage the conversion of source code into compiled code, but it can perform a variety of other tasks along the way.

I have two versions of MSBuild on my system:

```
C:\Windows\Microsoft.NET\Framework\v2.0.50727\MSBuild.exe
C:\Windows\Microsoft.NET\Framework\v3.5\MSBuild.exe
```

If you have at least one version of .NET installed, you should have at least one copy of MSBuild.exe. C7, to my knowledge, doesn't include MSBuild with the install, but since the IDE requires .NET anyway that isn't a problem.

Calling MSBuild

You can call MSBuild directly from a command prompt (i.e. a "DOS" window). In the following example I've navigated to a directory containing a C7 version of the DLLTutor sample application. You'll either have to have MSBuild.exe on the path or specify the full pathname, like this (your directory name may differ):

```
C:\Windows\Microsoft.NET\Framework\v2.0.50727\MSBuild.exe dlltutor.sln
```

Most likely, this command will result in a bunch of errors. On my machine these start off as:

```
error MSB4019: The imported project "C:\SoftVelocity.Build.Clarion.targets" was not found. Confirm that the path in
the <Import> declaration is correct, and that the file exists on disk.
```

The problem is this line in each of the cwproj files:

```
<Import Project="$(ClarionBinPath)\SoftVelocity.Build.Clarion.targets" />
```

SoftVelocity.Build.Clarion.targets is an XML file that tells MSBuild how to use the Clarion compiler. And there's no value set for ClarionBinPath, so MSBuild interprets the location of that file as C:\SoftVelocity.Build.Clarion.targets which clearly doesn't exist. You have to specify the location of that file, and you can do so either by setting an environment variable called ClarionBinPath (pointing to the C7 bin directory) or by passing the value on the command line, like this (line breaks added):

```
C:\Windows\Microsoft.NET\Framework\v2.0.50727\MSBuild.exe ↵
 dlltutor.sln /property:ClarionBinPath=↵
 "C:\Program Files\SoftVelocity\Clarion 7\bin"
```

This command compiles the solution in exactly the same way the IDE does.

Loggers, tasks, and infinite possibilities

The above command simply duplicates the C7 functionality. To add value you can add parameters and/or create a customized build file. I'll come back to customized build files toward the end of this article.

By using the appropriate option can validate your project file, insert command line settings from a file, specify which parts of a build should be done and in what order, set logging options and more. To see the command line help use the /h or / help parameter:

```
C:\Windows\Microsoft.NET\Framework\v2.0.50727\MSBuild.exe /h
```

You can find more information on the MSBuild command line options at MSDN (that's for the .NET 2.0 version used by C7). For now, I'll just consider the possibilites of the /logger option. Here's the help for /logger:

```
 /logger:<logger>

Use this logger to log events from MSBuild. To specify
multiple loggers, specify each logger separately.
The <logger> syntax is:
  [<logger class>,]<logger assembly>[;<logger parameters>]
The <logger class> syntax is:
  [<partial or full namespace>.]<logger class name>
The <logger assembly> syntax is:
  {<assembly name>[,<strong name>] | <assembly file>}
The <logger parameters> are optional, and are passed
to the logger exactly as you typed them. (Short form: /l)
Examples:
 /logger:XMLLogger,MyLogger,Version=1.0.2,Culture=neutral
 /logger:XMLLogger,C:\Loggers\MyLogger.dll;OutputAsHTML
```

MSBuild loggers are essentially DLLs that conform to certain specifications. You tell MSBuild to use an alternate logger via command line options. Loggers are .NET assemblies (DLLs, in fact).

When you're compiling in the IDE the default console logger is probably fine, but what about compiling outside the IDE? Let's say you've set up a dedicated machine, a build server, to handle the assembly of a large, multi-DLL solution. How do you know the compile was successful? You could walk over to the build machine and look at the monitor; you could log the results to a file and examine the file, or you could set up a custom logger to report build results whatever way you like. If, for instance, you have Clarion# it seems to me that it should be possible to create a logger to write results to a TPS file.

You can also customize MSBuild through the use of tasks. Like loggers, tasks are .NET assemblies, but instead of specifying them on the command line you specify them in the solution file. (Since the Clarion IDE modifies solution files, it's probably safest to create a custom version of your solution file and build that using a direct call to MSBuild.) The details of implementing tasks are beyond the scope of this article, but you can find plenty of information online. Start with the MSDN page on tasks, and then have a look at the MSBuild Wiki.

You don't necessarily have to write your own tasks. Check out the open source MSBuild tasks at Tigris. Tasks that might be useful to Clarion developers include:

- Playing sounds/beeps
- Deleting files
- Uploading files to an FTP server
- Installing/Uninstalling files in the GAC
- Sending email
- Running NUnit tests (with Clarion#)
- Reading/writing the registry
- Executing SQL
- Reading/writing XML
- Zipping files
- Controlling IIS
- Compiling HTML Help
- Compressing Javascript
- Access a Subversion repository

Let's say you're managing a team of Clarion# developers writing an ASP.NET application (the principles apply to C7 desktop development as well). You're using Subversion for version control, and your developers check in their updated source code on a regular basis. You could set up a scheduled nightly build that:

1. Checks out the latest sources
2. Compiles the code
3. Updates the test web server with the compiled code
4. Compresses any Javascript used by the web app
5. Reloads the test SQL database
6. Executes the unit tests
7. Zips up the application's files
8. Uploads the zip to an FTP server for beta testers
9. Emails the beta testers notifying them of a new release
10. Emails you at each stage with a status report

Let's see you do *that* with C6.

But there's more. The .NET 3.5 release of MSBuild has an option to take advantage of multicore CPUs, where each core handles a compile. There some problems with multicore compiles and C7, but under some circumstances it does work. Ben Dell and I will have more to say on this subject next month. Stay tuned....

Resources

- MSBuild on MSDN (.NET 2.0)
- MSBuild on MSDN (VS 2008 / .NET 3.5)
- MSBuild Wiki
- Also have a look at Icetips' Build Automator, which works with MSBuild and with C6.3 apps

---

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

Add a comment

Clarion Magazine

The Clarion Library

Clarion Magazine keeps an archive of past Clarion publications. This page lists publications we currently have on file. If you have any of the items listed as *not in library*, or have items in excellent condition which we list as Fair or Poor, please consider sending them to us. If you wish to be compensated for shipping costs, please contact us first to make arrangements.

Send publications to:

CoveComm Inc.
1036 McMillan Ave
Winnipeg, MB R3M 0V8
CANADA

The Clarion Call (Clarion Software's Newsletter)		
Issue	**Condition**	**Notes**
V1N1	Excellent	Summer 1987 (this may in fact be the same thing as V1N1 of the Clarion Developer, although the volume number doesn't seem right)
Clarion Developer (Clarion Software's Newsletter)		
Issue	**Condition**	**Notes**
V1N1	*not in library?*	See Clarion Call V1N1.
V1N2	*not in library*	
V1N3	*not in library*	
V1N4	*not in library*	
V2N1	*not in library*	
V2N2	*not in library*	
V2N3	*not in library*	

V2N4	*not in library*	
V3N1	*not in library*	
V3N2	*not in library*	
V3N3	*not in library*	
V3N4	*not in library*	
V4N1 Winter 1991	Good	
V4N2 Spring 1991	Fair	
V4N3 Summer 1991	Good	
V4N4 Fall 1991	Good	
V5N1 Winter 1992	Good	
V5N2 Spring 1992	Good	
[V5N3] 3rd Quarter 1992	Good	
[V5N4] 4th Quarter 1992	Good	
[V6N1] 1st Quarter 1993	Good	
subsequent issues?	*not in library*	
Volume 1, 1994	Good	
subsequent issues?	*not in library*	

Clarion Tech Journal

Issue	**Condition**	**Notes**
V1N1 1989	Fair	In spiral bound edition (V1N1-V3N1) only
V1N2 1989	Fair	In spiral bound edition (V1N1-V3N1) only
V1N3 1989	Fair	In spiral bound edition (V1N1-V3N1) only

V2N1 1990	Fair	In spiral bound edition (V1N1-V3N1) only
V2N2 1990	Fair	In spiral bound edition (V1N1-V3N1) only
V2N3 1990	Fair	In spiral bound edition (V1N1-V3N1) only
V2N4 1990	Good	
V2N5 1990	Good	
V2N6 1990	Good	
V3N1 1991	Good	
V3N2 1991	Good	
V3N3 1991	Good	
V3N4 1991	Good	
V3N5 1991	Good	
V3N6 1991	Good	
V4N1 1992	Good	
V4N2 1992	Good	
V4N3 1992	Good	
V4N4 1992	Good	
V4N5 1992	Good	
V4N6 1992	Good	
V5N1 1993	Good	
V5N2 1993	Good	
V5N3 1993	Good	
V5N4 1993	Good	
V5N5 1993	Good	
V5N6 1993	Good	

V6N1 1994	Good	
V6N2 1994	Good	
V6N3 1994	Good	
V6N4 1994	Good	
V6N5 1994	Good	
V6N6 1994	Good	
V7N1 1995	Good	
V7N2 1995	Good	
V7N3 1995	Good	
V7N4 1995	Good	

Clarion Developer's Forum

Issue	**Condition**	**Notes**
Announcement	Good	Subtitled "The NEW Clarion/ TopSpeed Users Journal - this is an unnumbered issue. Comb binding.
Volume 1, Number 1	Good	Dec 1995. Comb binding.
subsequent issues?		*Did this publication continue beyond V1N1?*

Clarion for Windows Journal

Issue	**Condition**	**Notes**
V1N1	Good	
V1N2	Good	
V1N3	Good	
V1N4	Good	
V1N5	Good	
V1N6	Good	

Reader Comments

Add a comment

Clarion Magazine

The ClarionMag Blog

Get automatic notification of new items! RSS feeds are available for:

XML All blog entries
XML All new items, including blogs

Blog Categories

- »All Blog Entries
- »Clarion 7 Clarion.NET
- »Future Articles
- »News flashes
- »Nifty Stuff

Clarion magazines and newsletters over the years

Direct link

Posted Thursday, February 26, 2009 by Dave Harms

There have been a number of Clarion-oriented print publications over the years, from newsletters to bound journals. The two best-known are Clarion Tech Journal and Clarion for Windows Journal. A few years back I realized I had most of the CTJ and CWJ issues, and a few old Clarion newsletters. So I decided to begin a collection.

Recently I got a couple of copies of something called the Clarion Developer's Forum, published by David King, from Randy Rogers. I don't recall ever hearing of this publication, and as far as I know the announcement issue and V1N1 are the only issues ever printed. And I've heard there were some UK newsletters as well, which I've never seen.

I'm missing most of the early issues of Clarion Software's newsletter. Larry Sand sent me an original 1987 color brochure and a copy of Volume 1, No 1 of the Clarion Call, which I presume later became the Clarion Developer newsletter (although the numbering doesn't seem to add up).

Have a look at the library page, and if you can fill in some of the gaps please let me know.

Marty's TPS to SQL templates

Direct link

Posted Tuesday, February 24, 2009 by Dave Harms

Marty Honea, one of ClarionMag's regular authors (every year, like clockwork<g>), sent me some TPS to SQL templates today. They mainly cover MS SQL, but include code for MySQL and Oracle as well. The templates generate the SQL script, TXDs with the new structure, and create a PRJ that converts the TPS files to the SQL database. You can find them on the ClarionMag download page.

CreateFileMapping mystery solved? (Updated)

Direct link

Posted Monday, February 23, 2009 by Dave Harms

This is an intermittent error - sometimes it goes away on the next build. But Oscar appears to have found the cause. After going through all the processes running on his development machine, he identified the ATI Catalyst Control Center (CCC.EXE) as the culprit. After killing that process he was able to compile without difficulty, and Abe Jimenez confirmed.

Abe advises running MSConfig and checking the Startup tab for a program called StartCCC.exe.

Naturally, you make any changes to your computer's configuration at your own risk. Please be careful! You might want to try killing the process from the Task Manager first; if everything still works as expected, then you can try disabling it at startup.

I've never encountered the CreateFileMapping problem (at least not that I can recall) even though I do have Catalyst Control Center on my machine. But my CCC is launched by CLIStart.exe, not by StartCCC.exe, so perhaps it's a version-specific problem.

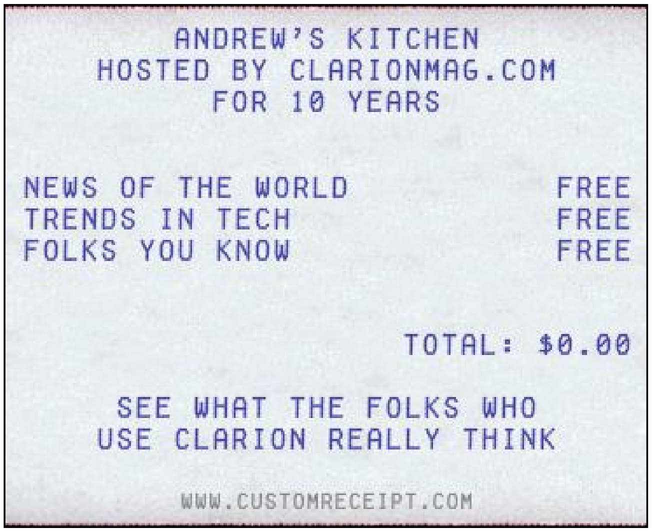
Update: I mixed up the credits - Abe pointed out MSConfig/StartCCC.

In the kitchen

Direct link

Posted Tuesday, February 10, 2009 by Dave Harms

Andrew Guidroz posted this nifty graphic in (where else) Andrew's Kitchen.



```
        ANDREW'S KITCHEN
    HOSTED BY CLARIONMAG.COM
          FOR 10 YEARS

NEWS OF THE WORLD           FREE
TRENDS IN TECH              FREE
FOLKS YOU KNOW              FREE

                    TOTAL: $0.00

     SEE WHAT THE FOLKS WHO
   USE CLARION REALLY THINK

       WWW.CUSTOMRECEIPT.COM
```

For all you subscribers who haven't been to the kitchen yet, the newsgroup server's at news.clarionmag.com and the user id and password were in your subscription confirmation email.