

Clarion Magazine

Clarion News

- » [List & Label August 15 Deadline](#)
- » [EasyOpenOffice 1.00](#)
- » [EasyCOMCreator 1.01](#)
- » [Clarion 6.2 Build 9047](#)
- » [FullRecord 1.00](#)
- » [EasyCOMCreator 1.00](#)
- » [iAlchemy's daFinga](#)
- » [ClaProtect Adds ASprotect to Clarion Apps](#)
- » [xDigitalClock 1.8](#)
- » [BST 4.112](#)
- » [EasyResizeAndSplit 2.10](#)
- » [EasyListPrint 1.13](#)

[\[More news\]](#)

Podcast



SoftVelocity president Bob Zaubere is back on Planet Clarion. Andrew and Dave talk with Bob about the ConDev AVIs and what they really mean, what's involved in creating the new IDE, and the differences between Win32 and .NET screen formatters. 47:00:00, 16913K

 [Listen now](#)

[\[Track lists, more podcasts\]](#)

Latest Free Content

[C6 Tips Book In Print, C5.x Tips Books Re-Released](#)

Clarion 6 Tips & Techniques Volume 1 is now in print, and although we've raised the price you can still save a few bucks off the list price for a limited time. We've also re-released the original tips book in a two volume set, Clarion 5.x Tips & Techniques Volume 1 and 2. These books are in print now, and for a very limited time are available for \$29.95 each to subscribers, and \$32.95 each to non-subscribers.

[Clarion 7/Clarion.NET News - Updated June 1, 2005](#)

Read our Clarion 7/Clarion.NET blog page for the latest news on SoftVelocity's upcoming product releases. In the latest edition, Bob Z responds to the discussion of the Brazilian DevCon presentations.

[PDF: Using Agile Programming Techniques for the Enterprise Information System : A Case Study](#)

Louis Coraggio and Wayne Lundeberg describe how Clarion is used in an extreme/agile programming environment to create an Enterprise Information System for an ISO 9001 manufacturing firm. An overview of the EIS development process, the system design goals, and a chronological narrative of EIS development are presented. Included are additional requirements and recommendations for those considering agile methods.

[\[More free articles\]](#)

**Save up to 50% off ebooks.
Subscription has its rewards.**



Latest Subscriber Content

[Planet Clarion Transcript: Bob Zaubere Part 2](#)

In Part 2 of the June podcast interview, Bob Zauere talks about COM vs .NET, C7 productivity gains, pricing, XML, Code DOM, and the stuff nobody has asked about yet. This part of the interview is only available in transcript form.

Posted Thursday, July 28, 2005

[Planet Clarion Transcript: Bob Zauere on the new IDE and Clarion.NET](#)

The unplanned release of the ConDev AVIs showing the new IDE created much confusion among Clarion developers. In this interview Bob Zauere explains the context of the AVIs, and discusses the C7 and Clarion.NET development roadmap.

Posted Friday, July 22, 2005

[C6 Tips Book In Print, C5.x Tips Books Re-Released \(free article\)](#)

Clarion 6 Tips & Techniques Volume 1 is now in print, and although we've raised the price you can still save a few bucks off the list price for a limited time. We've also re-released the original tips book in a two volume set, Clarion 5.x Tips & Techniques Volume 1 and 2. These books are in print now, and for a very limited time are available for \$29.95 each to subscribers, and \$32.95 each to non-subscribers.

Posted Friday, July 01, 2005

[Online Mapping And Routing In Clarion](#)

Online mapping is a useful and increasingly popular tool for personal and business use. Colin Wynn shows how easy it is to interface to MSN mapping and routing with a Clarion application.

Posted Wednesday, June 22, 2005

[A Multi-Threaded Queue Of Queues](#)

Queues make all kinds of data handling fast and easy. With Clarion 6, queue access gets a little more complicated, since queues by themselves are not thread safe. And what if you use queues of queues? Svetlana Zusman shows how to lock nested queues for thread safety and maximum efficiency.

Posted Tuesday, June 21, 2005

[Brazilian ConDev - Notes and Photos](#)

Clarion Magazine's undercover reporter gets the real story on this year's Brazilian conference...

Posted Monday, June 13, 2005

[Brazilian ConDev Report](#)

Fernando Cerini reports on the sixth annual Condev Mercosul Gescla, organized by the

Brazilian User Group.

Posted Monday, June 13, 2005

[PDF for May 2005](#)

All Clarion Magazine articles for May 2005 in PDF format.

Posted Wednesday, June 08, 2005

[Putting Clarion 6 Under Version Control With TortoiseSVN](#)

You might be using version control software for your own source code, but you should probably be using it for Clarion releases as well. As David Harms shows, this is easy to achieve with TortoiseSVN, Subversion, and a local repository.

Posted Wednesday, June 08, 2005

[Planet Clarion Transcript: Clarion 6.2 Released](#)

In this transcript of the May 25, 2005 show Andrew Guidroz and Dave Harms discuss the release of Clarion 6.2, its features and upgrade fee, the subscription program, and Clarion 7.

Posted Friday, June 03, 2005

[Clarion 7/Clarion.NET News - Updated June 1, 2005 \(free article\)](#)

Read our Clarion 7/Clarion.NET blog page for the latest news on SoftVelocity's upcoming product releases. In the latest edition, Bob Z responds to the discussion of the Brazilian DevCon presentations.

Posted Wednesday, June 01, 2005

[\[Last 10 articles\]](#) [\[Last 25 articles\]](#) [\[All content\]](#)

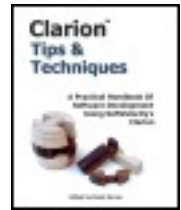
Printed Books & E-Books

[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 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

Clarion Magazine

Clarion News

[Search the news archive](#)

List & Label August 15 Deadline

All List & Label users should have already had an email from combit reminding you that the deadline is fast approaching to secure your discount of 20% off the latest version of List & Label. For more information about the new features of List & Label v11 (especially multi-child tables which has been a request for a long time), please visit <http://www.combit.net>. Solace Software is processing upgrade orders at the same price as Combit, plus if you order your upgrade through Solace you will get the upgrade templates to v11 which are being worked on at the moment.

Posted Thursday, August 11, 2005

EasyOpenOffice 1.00

EasyOpenOffice is a set of classes and templates allowing you to exchange data between Clarion applications and OpenOffice Calc and Writer. This release has basic features (you can open and close Calc and Writer, pass/get the data form your Clarion application, format cells in Calc, create tables and set font properties in Writer etc.). More features to come. Clarion 5.5 or Clarion 6.1/6.2. ABC, Legacy class template support. 32-bits only.

Price: \$149

Posted Wednesday, August 10, 2005

EasyCOMCreator 1.01

EasyCOMCreator 1.01 is now available. Changes include: Compilation fixes; Main window appearance; New ProgID and VersionIndependentProgID properties; C++ console test app example for the Math object.

Posted Wednesday, August 10, 2005

Clarion 6.2 Build 9047

SoftVelocity has released an update for Clarion 6.2. The new release is build 9047 and it corrects several recently reported problems. The installs are patches and must be applied against release 9046.

Posted Monday, August 08, 2005

FullRecord 1.00

FullRecord 1.00 is now available. This is a full-featured auditing system for your Clarion or ABC programs, 5.5 or 6.x, ABC or Legacy. Multi-DLL support, 100% source code. FullRecord can record operations from any type of file (ISAM, SQL, even IMDD if needed). The auditing file may be ISAM or SQL. DIM fields are supported (full 4 dimensions support); OVERed fields are supported; Unlimited memos support per file. There is only ONE file for auditing storage. BLOB recovering is supported for clarion 6.x. Available at ClarionShop US#99, or get the FullRecord + FinalStep bundle for US\$169.

Posted Monday, August 08, 2005

EasyCOMCreator 1.00

The EasyCOMCreator utility is used to create a Component Object Model (COM) dynamic-link library (DLL) automatically using Clarion. It allows you to manage your Projects, Objects, Methods and Parameters and generates all the necessary CLW/PRJ files (including a test application) automatically. After you compile your Project you will get your COM server DLL. To use it, you need to register it using the REGSVR32.EXE utility and you are done. Requires Windows 98 or higher, Clarion C61-9034 or higher. Demo version available (limited to 5 methods in all).

Posted Monday, August 08, 2005

iAlchemy's daFinga

iAlchemy has released daFinga, a template/library designed around the APC Biopod biometric fingerprint readers. ABC or Clarion template chain. Enroll multiple fingers per user, identify, list, delete. All functions are accessible either through a simplified control/extension templates or can be hand coded as simple functions. \$179.00 USD (electronic delivery).

Posted Monday, August 08, 2005

[ClaProtect Adds ASprotect to Clarion Apps](#)

ClaProtect is a set of templates and classes that allows you to use all features of ASprotect, a popular and powerful program protection system. ASprotect lets you encrypt important code, make keys limited by user, date, hardware (machine id), or function. ASprotect includes anti-debugger/CRC protection and packs executables. It uses 512/768/1024bit RSA keys.

Posted Monday, August 08, 2005

[xDigitalClock 1.8](#)

xDigitalClock 1.8 is now available, and features an easy-to-use class and control Template to display time like a digital clock. Can also be used as a digital stop watch and a countdown timer. Digital skins are supported. Five digital sets link into the program. For each xDigitalClock object you can set its own INI file for saving the configuration. Now compatible with Clarion 6.2.

Posted Monday, August 08, 2005

[BST 4.112](#)

Big Scheduler Tamer Template Suite 4.112 is now available. This release fixes some bad icon references in the demo, as well as a drag/drop problem in the demo.

Posted Monday, August 08, 2005

[EasyResizeAndSplit 2.10](#)

EasyResizeAndSplit 2.10 is now available. Includes a bug fix for a focus problem, and a new method to reset all ERS settings. Price: \$79 (1 license) or get a bundle of Easy3DStyle and EasyResizeAndSplit, individual cost \$168, for \$129. Free upgrade for all registered customers.

Posted Monday, August 08, 2005

[EasyListPrint 1.13](#)

EasyListPrint 1.13 is now available. Changes include a fix to the CSV icon to add it to the project, a fix to the browse box printing template, and a new ability to change the browse object name. This is version for Clarion 5.0, 5.5 and 6.1 (9034) / 6.2 (9046). Free for all registered customers. Purchase for \$69.

Posted Monday, August 08, 2005

[EasyCOM2INC 2.06](#)

EasyCOM2INC 2.06 has been released. This version includes fixes to "Load Defaults" behavior, and renames oleaut.lib to oleautcg.lib. The EasyCOM2INC utility is used to automatically creating Clarion include files with the definitions of COM-interfaces from IDL file, and now comes with EasyCOM Generator and generate needed Classes. Price: \$189. Free upgrade for all registered customers.

Posted Monday, August 08, 2005

[Clarioneers Database](#)

SealSoft has added about ten new Clarioners into its Clarioneers database. If you want to be listed in this database, send your information to SealSoft.

Posted Monday, August 08, 2005

[IceTips Products C6.2 Compatible](#)

All Icetips products should now be fully compatible with Clarion 6.2. Most of these builds do not contain any new features or updates, except that some of the products that called DLLs from templates are now using a single DLL written in C. Previously IceTips used close to 20 DLLs in its products for the various versions of Clarion. The Icetips Professional Wizards (<http://www.icetips.com/wizards-pro.php>) are the only product that is not using this new DLL, but it will be in the next release which will include a number of new features and additions.

Posted Monday, August 08, 2005

[xXPpopup 1.4](#)

New in xXPpopup 1.4: Compatible with Clarion 6.2; Small bugs were fixed in classes and template; New Installation kits created by SetupBuilder 5.0.

Posted Monday, August 08, 2005

[List & Label 11 Upgrades](#)

Combit have announced version 11 of List & Label which will be available around 15th August. As usual, Solace Software will supply upgrades which will include a free upgrade to version 11 of the templates. The main new features are: Multi-table/Sub reports support; Simplified Designer; Page 1/n support without multi passes; New format editor in the designer; New Frame editor; Rotational Header Lines in tables; Bulking changing of

common properties; Much better online help; New Aztec 2D Barcode; New aggregate functions rather than having to use sum variables; New toolbar controls on previewer and better range printing.

Posted Monday, August 08, 2005

[**xPictureBrowse 2.4**](#)

xPictureBrowse 2.4 is now available. New in this version: Compatible with Clarion 6.2; New Installation kits created by SetupBuilder 5.0; Some small bugs in class were fixed.

Posted Monday, August 08, 2005

[**xRuntimeStyle Manager 2.0**](#)

xRuntimeStyle Manager 2.0 has been released. New in this version: Compatible with Clarion 6.2; New installation kits created by SetupBuilder 5.0.

Posted Monday, August 08, 2005

[**Free xFunction Library 2.2**](#)

New in this release of the free xFunction library: Compatible with Clarion 6.2; New installation kit created by SetupBuilder 5.0; New date functions; new xGetFileVersion function.

Posted Monday, August 08, 2005

[**xWhatsNew Class 1.9**](#)

xWhatsNew Class 1.9 is now available. New in this version: Now compatible with Clarion 6.2; New Installation kits created by SetupBuilder 5.0.

Posted Monday, August 08, 2005

[**xPathManager 1.4**](#)

xPathManager 1.4 is now available. New in this version: Now compatible with Clarion 6.2; New Installation kits created by SetupBuilder 5.0.

Posted Monday, August 08, 2005

[**xInactivity 1.4**](#)

New in this xInactivity 1.4: Now compatible with Clarion 6.2; New Installation kits

created by SetupBuilder 5.0.
Posted Monday, August 08, 2005

[xFText 2.6](#)

xFText 2.6 is now available. New in this version: Now compatible with Clarion 6.2; New Installation kits created by SetupBuilder 5.0.
Posted Monday, August 08, 2005

[DOS Printer 10.31](#)

DOS Printer 10.31 has been released. DOS Printer is of interest primarily to CPD2.1 developers as it allows DOS software to print to any windows device without any coding changes. In Windows Xp or NT or 2000 , DOS Printer installs a special printer and is able to steal jobs from the print queue and print them using a CPCS report to any windows device. In Windows 9x, DOS Printer creates a desktop shortcut which uses the prn2fil program to redirect printer output to DOS Printer. DOS Printer is compiled in C6.2 and uses CPCS, Vivid Help's EmailReport, Handy Tools, WinEvent, and Tracker 's PDF Tools. Setupbuilder is used for the installation, and Armadillo/Esellerate for distribution. DOS Printer has now been sold into more than 20 countries.
Posted Monday, August 08, 2005

[xAppWallpaper Manager 2.3](#)

xAppWallpaper Manager 2.3 has been released. New in this version: Now compatible with Clarion 6.2; New Installation kits created by SetupBuilder 5.0.
Posted Monday, August 08, 2005

[xAppWallpaper 1.8](#)

xAppWallpaper 1.8 has been released. New in this version: Now compatible with Clarion 6.2; New Installation kits created by SetupBuilder 5.0.
Posted Monday, August 08, 2005

[MAV Direct ODBC 010 and 008 Release](#)

Mav Direct Library version 0.10 and Template version 008 have been released. The Library version has been tested with the Ingres r3 SQL server, and has some new features. The template version also has some improvements and new features.

Posted Monday, August 08, 2005

[dpQuery 2.05](#)

dpQuery 2.05 has been released (full and demo versions). Changes include a fix to "silent mode" and Clarion 6.2 9046 support. dpQuery includes libraries and a template that adds a powerful interactive resource (Query wizard) for importing of practically any external data into your program. Clarion 5.5 or Clarion 6.1, ABC, Legacy class template support, 32-bits only. Price: \$149 (demo available).

Posted Monday, August 08, 2005

[BoTran 2.2](#)

BoTran 2.2 is now available. This release includes a bugfix to multi-DLL mode for ABC, and some translation fixes.

Posted Monday, August 08, 2005

Clarion Magazine

Clarion Magazine's Podcasts

What's a Planet Clarion Podcast, you ask? In short, it's an internet radio show for Clarion developers. Basically we're talking about audio programming in MP3 format, which means that all you really need to do is click on one of the links below to download and listen to Planet Clarion. The term [podcasting](#) was coined to describe the download of these kinds of programs to the Apple iPod, but you don't need an iPod to listen to Planet Clarion. Just click on the links below.



If you want to automatically download the feed instead of coming to this web page and clicking on the links, try some of the RSS software listed at iPodder.org. Point the RSS reader of your choice at our Planet Clarion RSS feed:

XML <http://www.clarionmag.com/planetclarion.rss>

Comments? Send us an [email](#)

Planet Clarion is hosted by Dave Harms, Clarion Magazine's editor, and Andrew Guidroz II, your favorite Cajun.

Planet Clarion for June 30, 2005

Track	Start	Length	Size	Description
Download or stream (Free Access)	00:00:00	47:00:00	16913 K	SoftVelocity president Bob Zaunere is back on Planet Clarion. Andrew and Dave talk with Bob about the ConDev AVIs and what they really mean, what's involved in creating the new IDE, and the differences between Win32 and .NET screen formatters.

Individual tracks				
Track 1 S	00:00:00	00:01:53	664K	Intro
Track 2 S	00:01:53	00:24:16	8535K	Z discusses the inner workings of the new IDE, and the missing context of the ConDev AVIs (which were never intended for public release)
Track 3 S	00:26:09	00:11:41	4113K	Window controls under Win32 vs .NET, and a controversial statement or two about the Clarion.NET window structure.
Track 4 S	00:37:51	00:09:07	3208K	Andrew and Dave mull over Z's comments, and Dave talks about the new Clarion books and ClarionMag's summer schedule.

[\[Last 5 podcasts\]](#) [\[All podcasts\]](#)

Freedom to distribute podcasts

You are free to distribute *public access* podcasts from Clarion Magazine, provided you do not modify those podcasts, and you do not charge any fees for the podcasts. In other words, if you want to put a podcast up on your server, feel free.

You may not, however, distribute individual tracks without express permission from Clarion Magazine.

Production notes and Skype suggestions

A number of listeners have asked how we record Planet Clarion. While all anyone really needs to record a podcast is a microphone, a PC, and some recording software, our setup is a little bit more complicated, mainly because we (Andrew and Dave) live 1700 miles apart. It's all made possible by [Skype](#), a free Internet phone service, which we highly recommend. The human ear can detect sounds in the frequency range of about 50 Hz (cycles per second) to 20,000 Hz; Skype transmits frequencies from 50 to 8000 Hz, which is pretty much the range of human speech, as compared to the plain old telephone service (POTS) which has a relatively narrow range of 300 Hz to 3300 Hz.

Dave records the Skype conversation on his PC, in two tracks, using [Sony Vegas](#); one track is Dave's microphone, and the other is Andrew via Skype. We originally intended to

record Andrew's microphone on his PC as well, but the Skype recording was good enough that we just ran with that. The two-track approach makes it possible to compensate for differences in microphones, sound quality, and sound levels. Among other tools, Dave uses the [Endorphin](#) plugin (via the [VST-DX Wrapper](#), since Vegas does not support VST plugins natively).

We both use headsets rather than standalone microphones. Dave is geeked up with an AKG HS200, and Andrew uses an Altec Lansing AHS 502, which is a reasonably-priced, good quality headset. We recommend the Altec Lansing products for anyone wanting to get started with Skype. If you're not sure which is the best Altec Lansing (or other manufacturer's) model in your area, find out what the local gamers like to use, and you'll probably be in good shape. Make sure whatever you buy will work with your sound card - if you want to go high end, it's more likely that your headset will need a pre-amp between you and the sound card. Also while you can use Skype with a microphone and speakers, we don't recommend that setup. It's too easy for the microphone to pick up sound from the speakers. You may think everything's fine, but the person at the other end will hear an echo. A headset removes this problem, and it also keeps the microphone a constant distance from your mouth, giving you more freedom of movement (especially if it's a wireless headset).

If you are considering a USB headset, keep in mind that these sometimes have greater latency than a regular headset/sound card combination. There's always some delay between the time a signal is generated by the microphone and the time the hardware finishes processing, and of course it also takes time to transmit the signal to the person at the other end of your Skype conversation. You want to keep latency to a minimum.

Clarion Magazine

Clarion 7/Clarion.NET News - Updated June 1, 2005

by **David Harms**

Published 2005-06-01

Bookmark this blog-style page to keep up with the latest news on the upcoming releases of Clarion 7 and Clarion.NET.

Bob Z Responds

June 1, 2005

There's been a lot of discussion about Clarion.NET and C7 following the demonstration at the Brazilian DevCon. That demo showed the new IDE, minus the AppGen and Dictionary Editor, and focused on Clarion.NET rather than on C7. This has given rise to some wild speculation on what is and isn't part of the new IDE, and on the differences between Clarion 7 and Clarion.NET. I had a chance to talk with Bob Z about these issues, and we'll also have a full podcast with Bob in the near future. Meanwhile, here are a few clarifications.

Although the DevCon demonstration didn't show any C7 compiles, Bob had planned to, as this capability is already there. In fact, there are at present two internal versions of the IDE, one used for testing components, and one with the AppGen. The testing version was demonstrated with Clarion.NET code, but Bob had planned to show Win32 code compiled using Clarion7, as well as the ability to choose the version of Clarion to compile with (4,5,6,7) at the project level. Work on the AppGen is not complete, but in that other internal version it does exist in the IDE. AppGen is not simply being ported, it is being re-engineered for improved performance and new and improved functionality. This work is well along.

The Clarion IDE and the SharpDevelop IDE have been grafted together - and right now the surface appearance is the same, but the UI is going to change as well. SoftVelocity has a commercial code license for SharpDevelop, but the IDE shown is a hybrid of Clarion's IDE and key components of the SharpDevelop IDE. There are significant differences between the Clarion IDE shown and the free SharpDevelop IDE, and that will be more evident when AppGen and the DCT editor, and other new components, are integrated. SoftVelocity has a large IDE code base, and big chunks of this are used in the new IDE as well.

SharpDevelop, according to Z, is an excellent basis for implementing support for the Clarion language under .Net, and the SharpDevelop developers are very talented. But despite the cosmetic similarity the new Clarion IDE is quite a bit different under the hood from the Sharpdev IDE. "Let's not forget that the heart and soul of a Clarion IDE is AppGen and the DCT editor, and that will always be true."

As for product releases, Clarion 7 is much closer than Clarion.NET, at least the full release. There is synergy between the two products - for instance, Clarion.NET can (and does, as one option) use the C7 file drivers. "Bottom line, there are two teams on two different projects...so both could be released pretty close together, or we could decide to do a hand coders .Net and a Win32...or we could wait and do the full product. [It's] likely that CSP EE users will have an option to do some beta tests either way."

First Look At New Clarion IDE

May 31, 2005

Attendees at the VI Condev Mercosul Gescla conference were the first outsiders to see the new Clarion IDE. Bob Zauhere presented an overview in English, and Diego Borojovich discussed the new language features of Clarion.NET, in Spanish. These presentations are available for [download](#) from Clarion Magazine.

Although not yet confirmed by SoftVelocity, the new IDE is quite clearly based on [SharpDevelop](#), an open source IDE for C# and VB.NET projects. Since SharpDevelop is available as an open source IDE under the GNU Public License (GPL), there has been some question as to whether this would make the Clarion IDE open source. This is highly unlikely. A SharpDevelop developer posted the following in one of the SharpDevelop

forums:

It is possible to release Open Source software under several licenses at the same time, including commercial licenses - look at MySQL, sendmail or OpenOffice for well known examples. This can be done only when the rights owners all agree. The #develop rights are all in one hand so we can do this.

Evidently SoftVelocity has obtained a commercial license to the SharpSoft code, which would entitle them to modify it as needed without having to release the IDE as open source. Certainly licensing an existing product allows SoftVelocity to take advantage of a slick, contemporary IDE while focusing on implementing Clarion.NET and the new AppGen environment.

AppGen implemented, not yet integrated

For the most part, the new IDE looks like the open source version of [SharpDevelop](#). This is because the AppGen and Dictionary Editor components, although implemented, have not yet been integrated into the IDE. Hopefully a hand-coder's release will be available soon, as happened in Clarion for Windows 1.0, where the first release didn't support the AppGen either. There are some new navigation features, and there is of course color syntax highlighting for Clarion code. Intellisense wasn't demonstrated, at least that I could see, but it is one of the promised features.

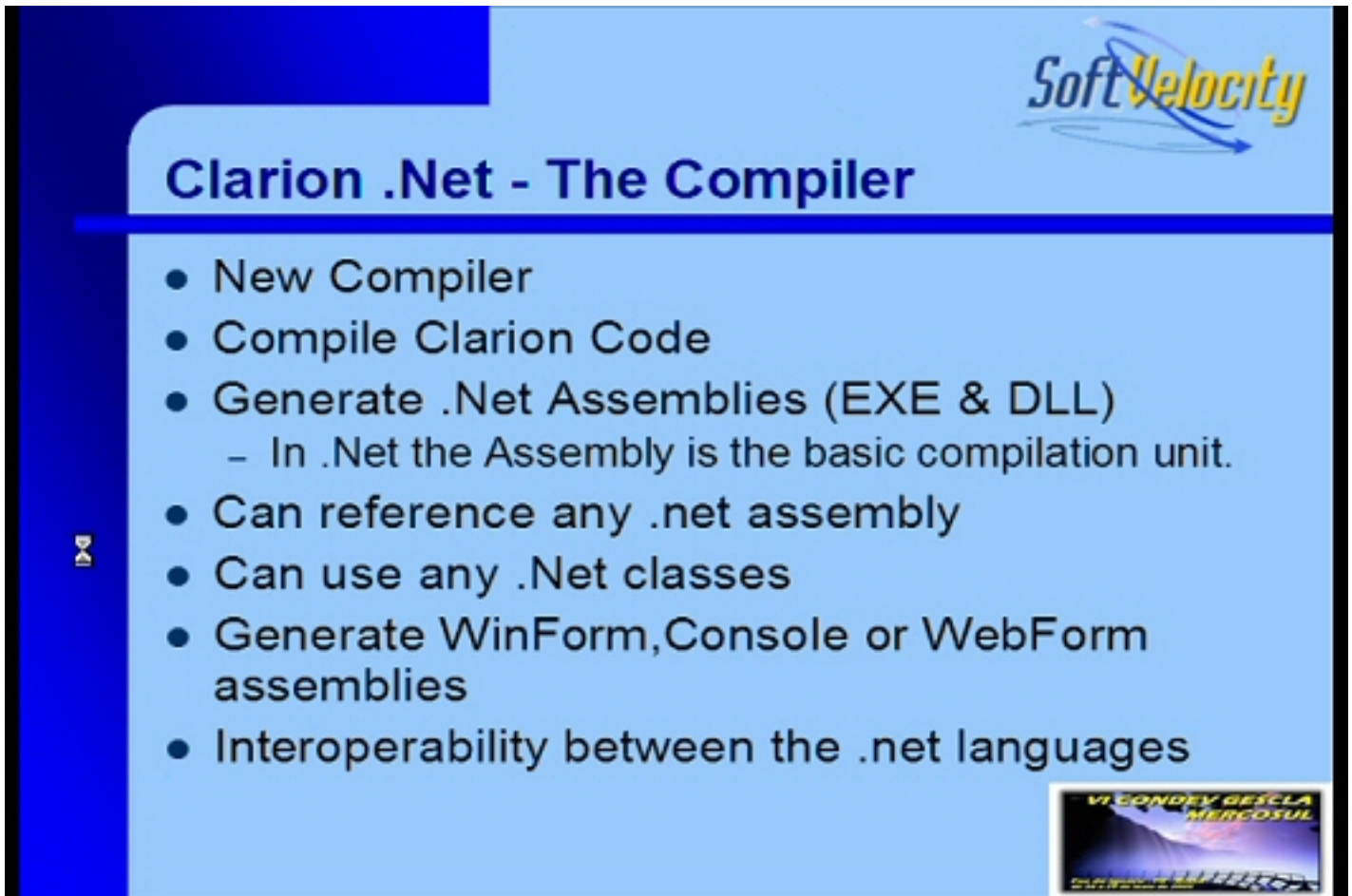
Download the AVIs

Bob and Diego's conference presentations were to be given live, but Bob's visa was delayed at the Brazilian embassy and he was unable to make the trip. Bob and Diego recorded their presentations on very short notice (note Bob's comments at the beginning of his session) so they are not as polished as you would otherwise expect.

- Bob Zaubere on the new IDE, high resolution, 11 minutes, **42 MB** ([download](#))
- Bob's PowerPoint presentation ([download](#))
- Diego Borojovich on Clarion.NET, medium resolution, 37 minutes, **150 MB** ([download](#))
- Camtasia codec - get this if you can't see the video ([download](#)).

Selected slides

The following screen shots are taken from Diego's presentation. Although the slides are in English, Diego's commentary is all in Spanish, and since I don't speak Spanish, much of what follows is idle speculation on my part.



SoftVelocity

Clarion .Net - The Compiler

- New Compiler
- Compile Clarion Code
- Generate .Net Assemblies (EXE & DLL)
 - In .Net the Assembly is the basic compilation unit.
- Can reference any .net assembly
- Can use any .Net classes
- Generate WinForm, Console or WebForm assemblies
- Interoperability between the .net languages

VI CONDEY BESLA
MERCOTUL

At least at this stage, it appears that Clarion.NET uses WinForms, as opposed to Clarion window structures. It's not clear yet what that means for porting applications from Win32 Clarion to Clarion.NET, although a direct port is not something that's ever been promised by SoftVelocity, at least as far as I know.



Clarion .Net - The Language

- Same Syntax than Clarion Win32 (C62,C55,etc.)
- New Syntax
 - Compatible with .Net, use of any .net class
 - Similar to C#



Clarion.NET does bring new capabilities, and that means some new language structures. These include some new class syntax, structured exception handling, delegates, and events.



Clarion .Net - The Syntax

A screenshot of the Clarion IDE. The top window shows two tabs: "Main.clw*" and "Form1.clw*". Below the tabs is a dropdown menu set to "Global Functions and Variables" and a variable declaration window showing "MyVarGlobal". The main code editor displays the following code:

```
1  PROGRAM
2
3
4
5
6
7
8  INCLUDE (' ERRORS . CLW ' ), ONCE
9  INCLUDE (' KEYCODES . CLW ' ), ONCE
10 !Declare Global Data
11 MyVarGlobal      STRING (30)
12
13 MAP
14 END
15
16 CODE
17     !Execute my code
18
```

The above slide shows the beginning of a simple Clarion program, as we now know it. The next slide shows the same program updated to Clarion.NET. The same basic structure remains, although there is a `NAMESPACE` directive for the current application, `USING` statements to pull in required .NET capabilities, and a change in the `STRING` variable which is now a `CLASTRING`. There has been some discussion of `CLASTRING`s already. In .NET, strings are classes, not simple data types, and `CLASTRING` is just a class wrapper for the Clarion `STRING` data type.



Clarion .Net - The Syntax

```

Main.clw* Form1.clw*
Global Functions and Variables MyVarGlobal
1  PROGRAM
2
3  NAMESPACE ('MDI')
4  USING ('System')
5  USING ('System.Drawing')
6  USING ('System.Windows.Forms')
7
8  INCLUDE ('ERRORS.CLW'), ONCE
9  INCLUDE ('KEYCODES.CLW'), ONCE
10 !Declare Global Data
11 MyVarGlobal    CLASTRING (30)
12
13 MAP
14 END           ⌚
15
16 CODE
17     !Execute my code
18     Application.Run (NEW Form1 ())
19

```

Another change is the use of `Application.Run()` to start the application. .NET is an object-oriented framework, and Clarion is a hybrid language with procedural roots and OOP extensions. It appears that in Clarion.NET, the procedural component is even shorter than in ABC and just gets the application loaded, sort of like a static `main()` method in a class.



Clarion .Net - The Syntax

```

Main.clw* Form1.clw*
Global Functions and Variables MyVarGlobal
1  PROGRAM
2
3  NAMESPACE ('MDI')
4  USING ('System')
5  USING ('System.Drawing')
6  USING ('System.Windows.Forms')
7
8  INCLUDE ('ERRORS.CLW'), ONCE
9  INCLUDE ('KEYCODES.CLW'), ONCE
10 !Declare Global Data
11 MyVarGlobal CLASTRING (30)
12
13 MAP
14 END
15
16 CODE
17 !Execute my code
18 Application.Run (NEW Form1 ())

```




Clarion .Net - The Syntax

- New Syntax
 - NAMESPACE
 - USING
- CLASS – extended syntax
 - Constructors with parameters
 - Properties (GET / SET)
 - Indexer ([] GET / SET)
 - Delegates (.Net function pointers)
 - Events (special type of Delegates)




I'm happy to see constructors being allowed parameters - that's not an answer to every initialization problem, but it's nice to have the option.



CLASS – extended syntax

Indexer ([] GET / SET)

```
IndexerClass CLASS, TYPE  
MYARRAY LONG, DIM(100)  
INDEXER(LONG index), LONG ! Indexer declaration  
END
```





CLASS – extended syntax

Indexer ([] GET / SET)

```

! Implementation of the indexer's getter/setter
! Getter implementation
IndexerClass.GET_INDEXER  PROCEDURE (LONG index)
    CODE
        ! Check the index limits.
        IF (index < 0 OR index >= 100)
            return 0;
        ELSE
            return SELF.MYARRAY[index];
        END

! Setter implementation
IndexerClass.SET_INDEXER  PROCEDURE (LONG index, LONG value)
    CODE
        IF (NOT (index < 0 OR index >= 100))
            SELF.MYARRAY[index] = value;
        END
  
```




Clarion .Net - The Syntax

- New Data Types
 - **ANY .NET DATA TYPE**
 - ADO.Net, XML, Winforms, Webforms, SQL.Data
 - 3rd party products (.net compatibles)
- Interoperability??????
 - Use of any .net assembly in your program
 - One DLL in Clarion.Net, the other in C# and one in VB.Net, and your EXE in Clarion.Net





Clarion .Net - The Syntax

- New Functionality
 - Exceptions (THROW / TRY / CATCH / FINALLY)
 - FOREACH (loop)
 - Supporting QUEUE
 - Arrays, etc, etc. 

- DataBinding
 - QUEUE



The new `ForEach` syntax lets you iterate through collections of objects. Basically it's another way of looping that knows how many objects there are to loop through, without you having to test for the end of the list.



Clarion .Net - FOREACH

```

NAMESPACE (' ShowForEach1 ')
PROGRAM
  map
  end

Ldim LONG, DIM(5)
L LONG

CODE
  Ldim[1] = 11
  Ldim[2] = 12
  Ldim[3] = 13
  Ldim[4] = 14
  Ldim[5] = 15

  FOREACH L IN Ldim
    MESSAGE (L)
  END

```



Clarion .Net - FOREACH

```

NAMESPACE (' ShowForEach1 ')
PROGRAM
  using (' System' )
  map
  end

  MyG GROUP
  {
    Id Int32
  }
  END

  MyQ QUEUE (MyG)
  END

  I MyG
  index long

  CODE
    LOOP Index=1 to 3
      MyQ.Id = Index
      ADD (MyQ)
    END

    FOREACH I IN MyQ
      MESSAGE (I.Id)
    END

```





Clarion .Net - Exceptions

```

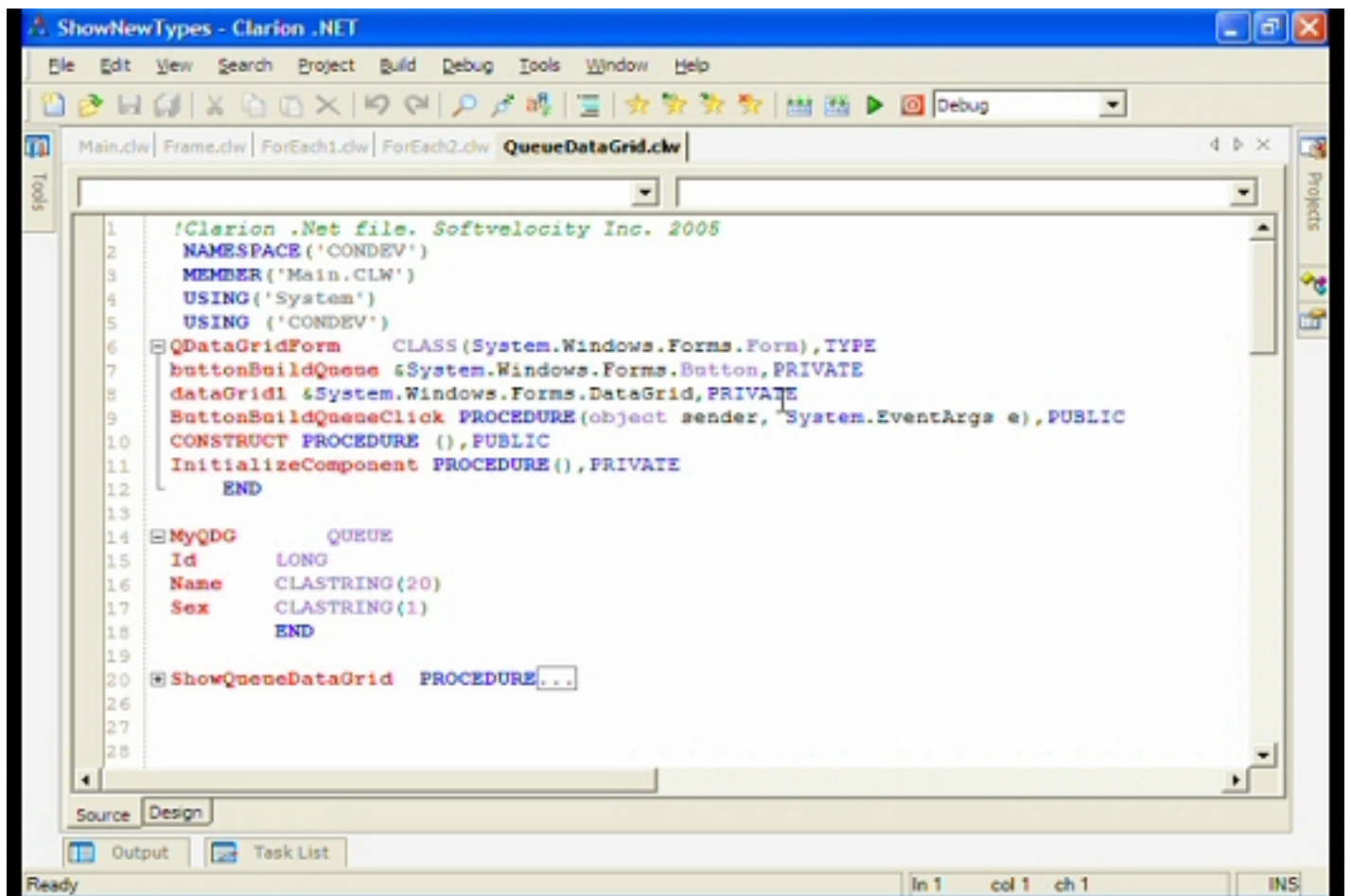
16
17 | TRY
18     IF 23 > '123'
19         LLL = 'Mayor'
20     ELSE
21         THROW new System.OutOfMemoryException()
22     END
23 CATCH (System.OutOfMemoryException)
24     LLL = 'Ocurrio una exception de memoria'
25 CATCH (System.Exception oEx)
26     LLL = 'Ocurrio una exception'
27 FINALLY
28     LLL = 'Todo Legal'
29 END

```

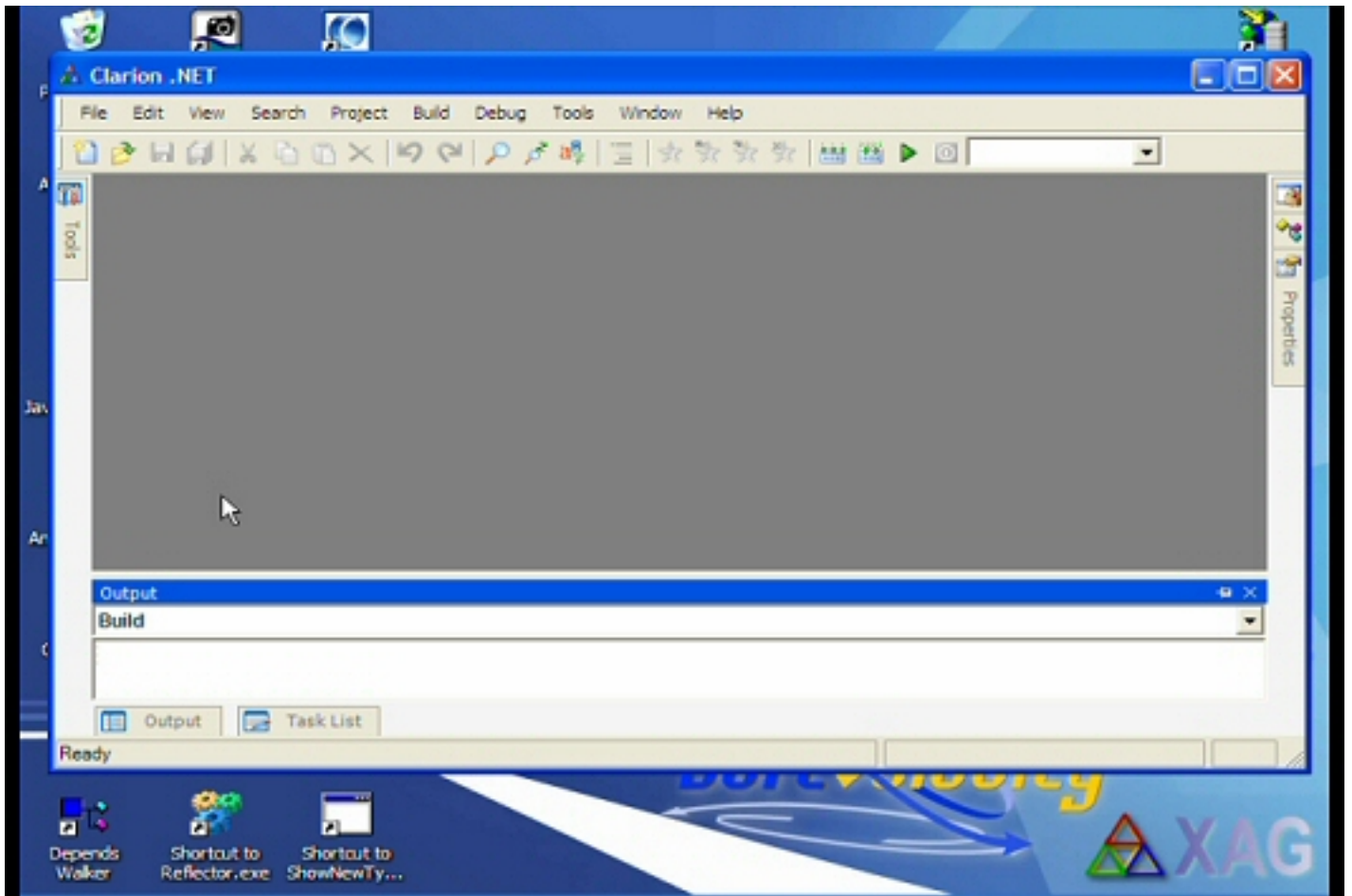


Exception handling is a very cool feature, and a much more flexible and powerful way of handling error conditions than simply testing for errors. Essentially you have a block of code marked with a TRY keyword - if any code in that block "throws" an error, none of the following code in the block executes - instead the code kicks straight through to one or more CATCH blocks which can test for specific errors.

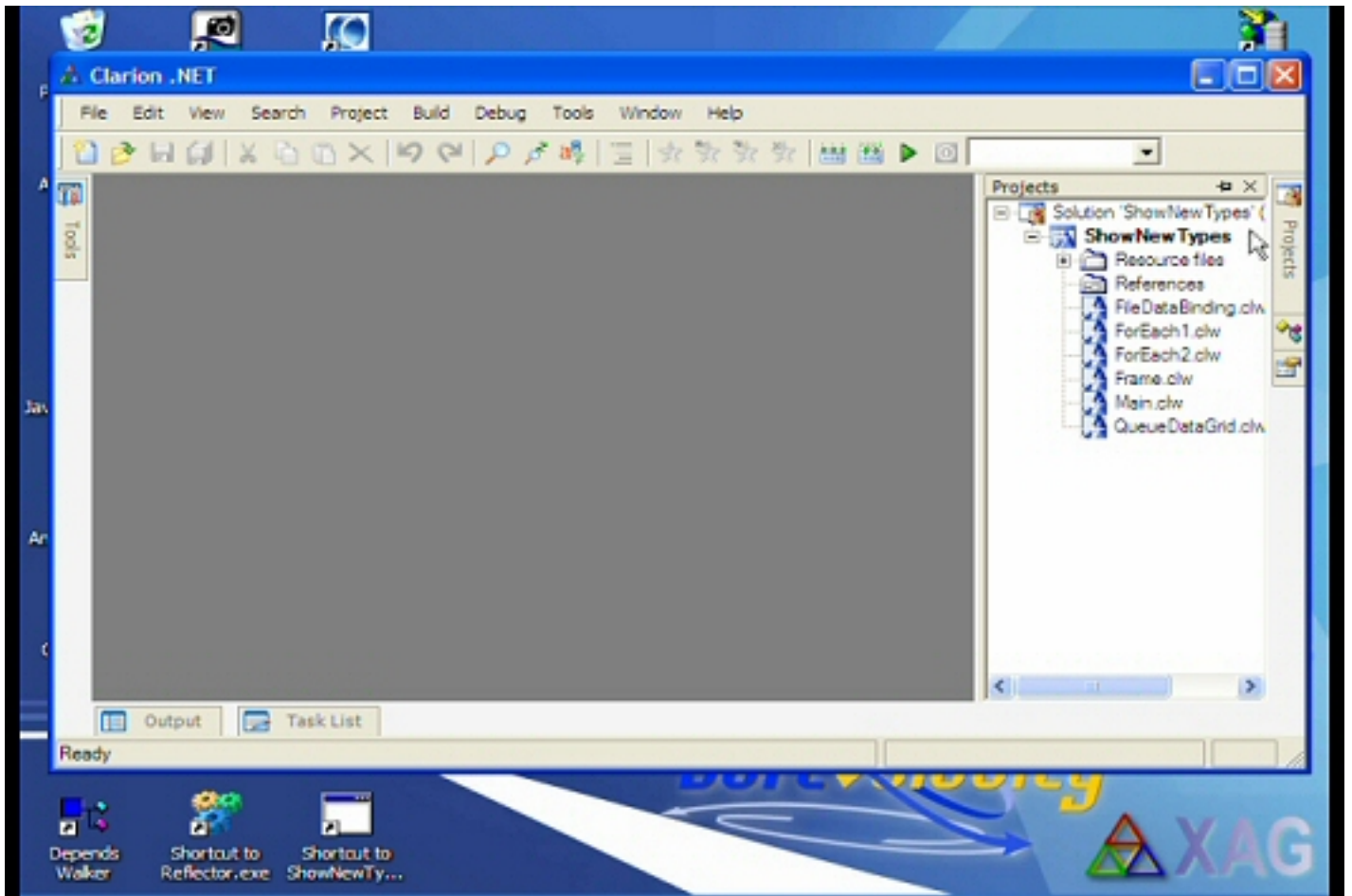
Here are a few more looks at the new IDE. I left these in for two reasons - first, I was unable to get a clean screen shot from Bob's presentation due to the higher resolution, and two, they show how configurable the IDE is. Bob had the project list on the left, while Diego has his sitting on a button on the right, where Bob had the property list. Also here the output window is docked, while Bob's was floating. And Diego generally has a lot less stuff visible on his screen - by comparison, take a look at the SharpDevelop [screen image](#).



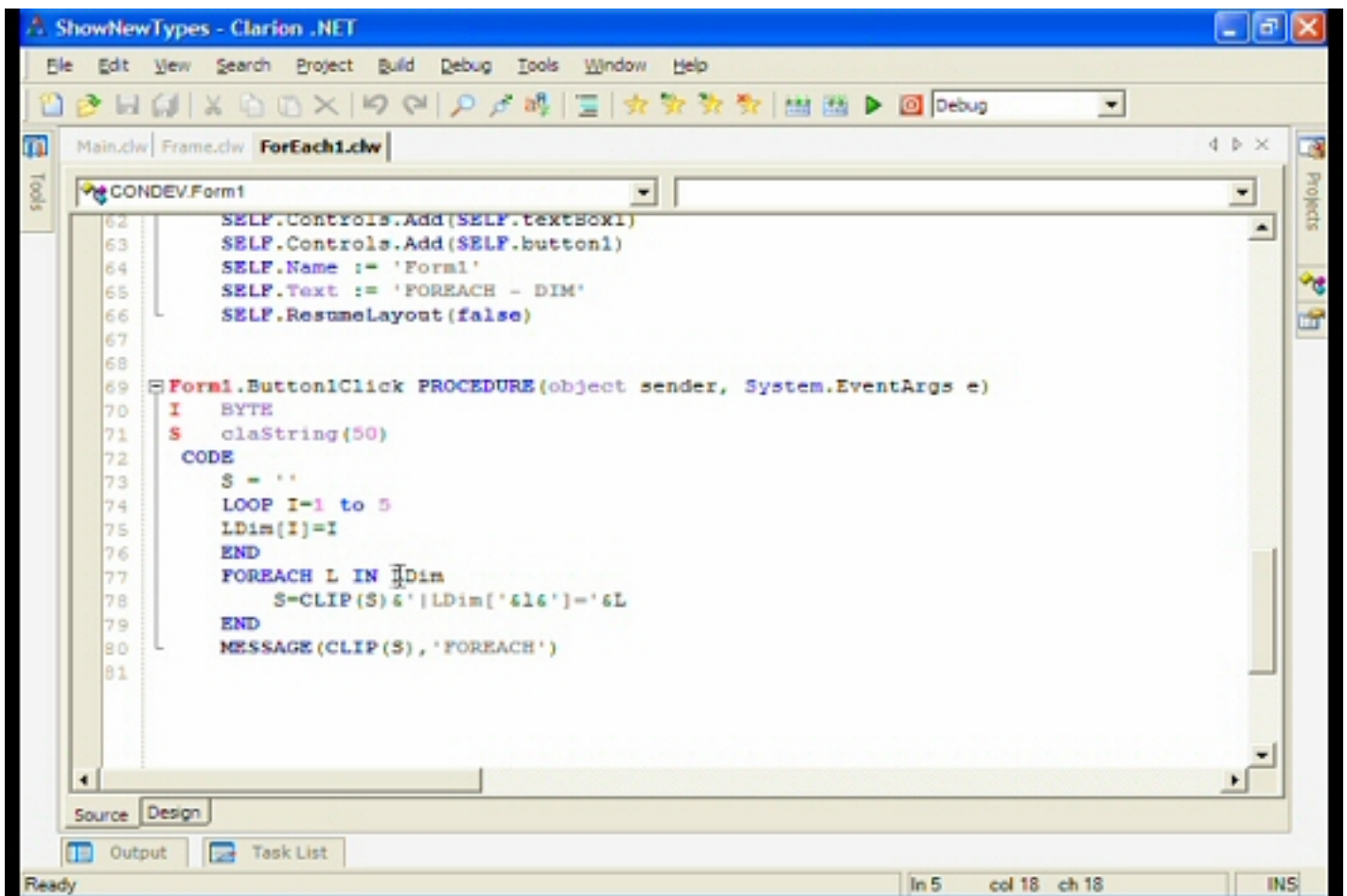
Displaying source code.



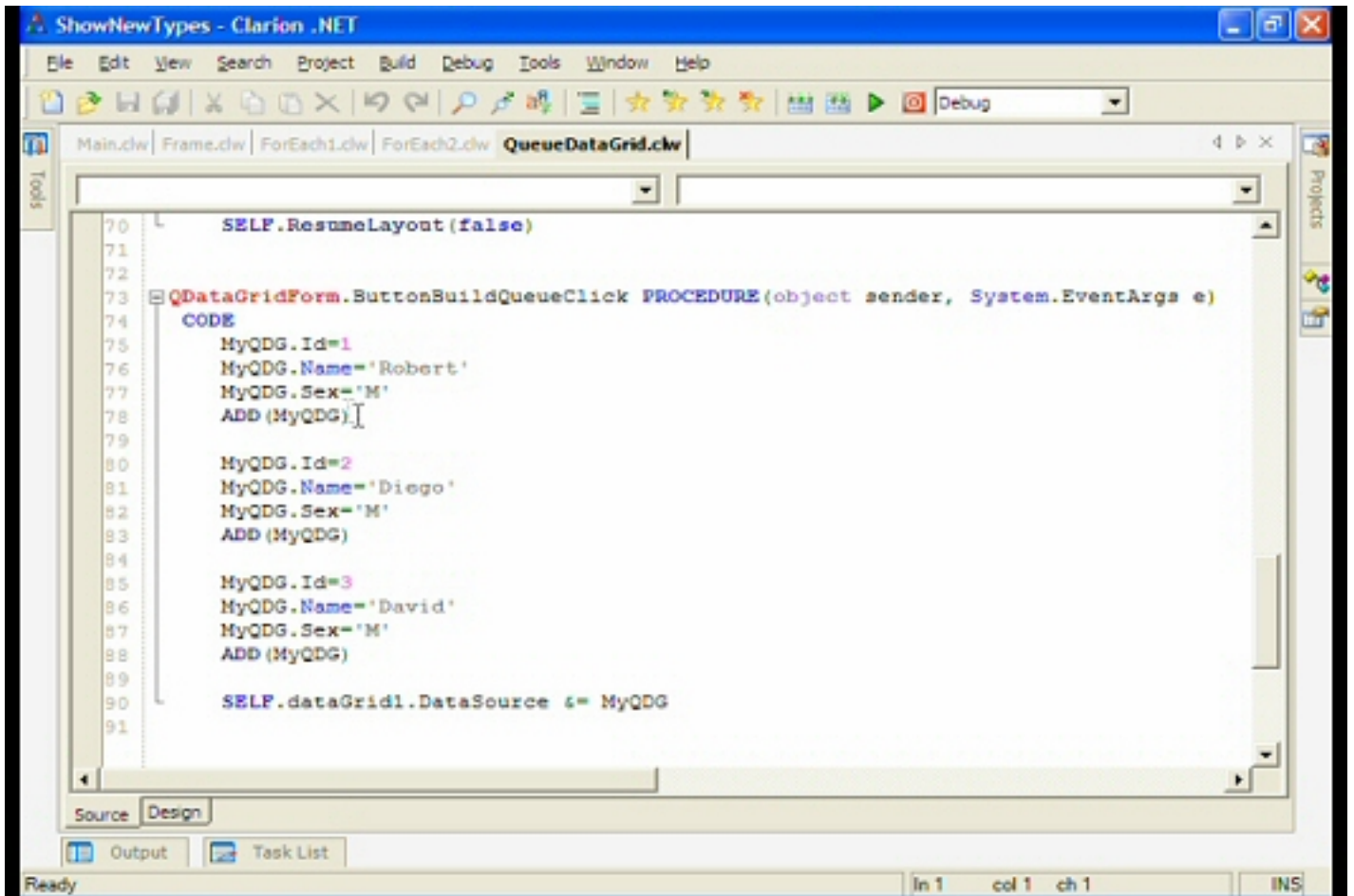
Note the docked Output pane.



Project menu on the right.

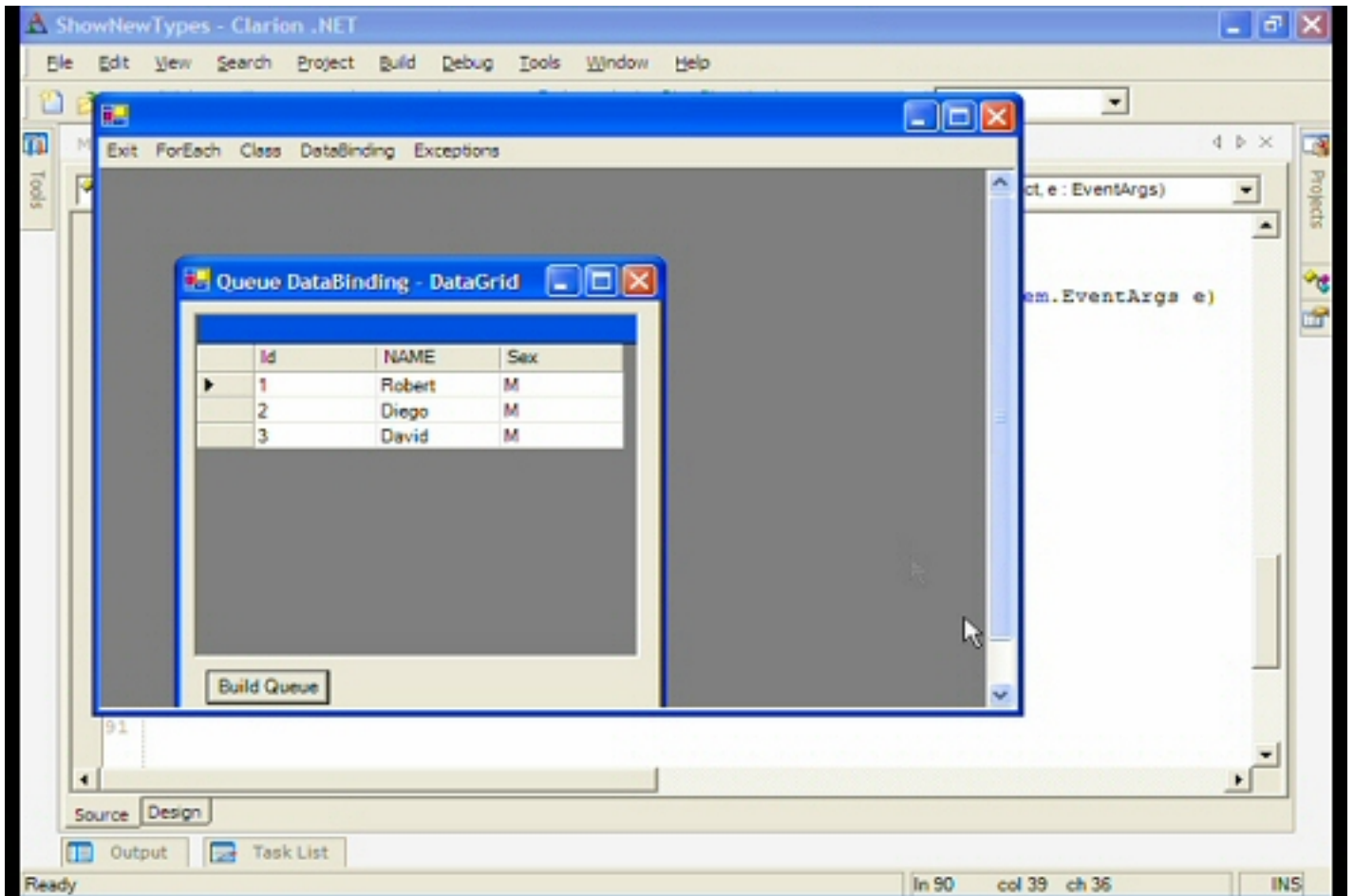


The above screen shot shows a button click handler, indicating a more mainstream approach to event handling. Clarion developers are accustomed to the `Accept` loop. Whether there is such an animal in Clarion.NET remains to be seen. This may not be a big issue since ABC developers at least are already familiar with embedding code in control `TakeEvent` embed points, rather than writing big blocks of code right into the `Accept` loop.



```
70     SELF.ResumeLayout(false)
71
72
73     QDataGridForm.ButtonBuildQueueClick PROCEDURE(object sender, System.EventArgs e)
74     CODE
75         MyQDG.Id=1
76         MyQDG.Name='Robert'
77         MyQDG.Sex='M'
78         ADD(MyQDG)
79
80         MyQDG.Id=2
81         MyQDG.Name='Diego'
82         MyQDG.Sex='M'
83         ADD(MyQDG)
84
85         MyQDG.Id=3
86         MyQDG.Name='David'
87         MyQDG.Sex='M'
88         ADD(MyQDG)
89
90     SELF.dataGrid1.DataSource &= MyQDG
91
```

Here's some code to build data to be displayed in a queue. Note the last line, which attaches the queue to the data grid.



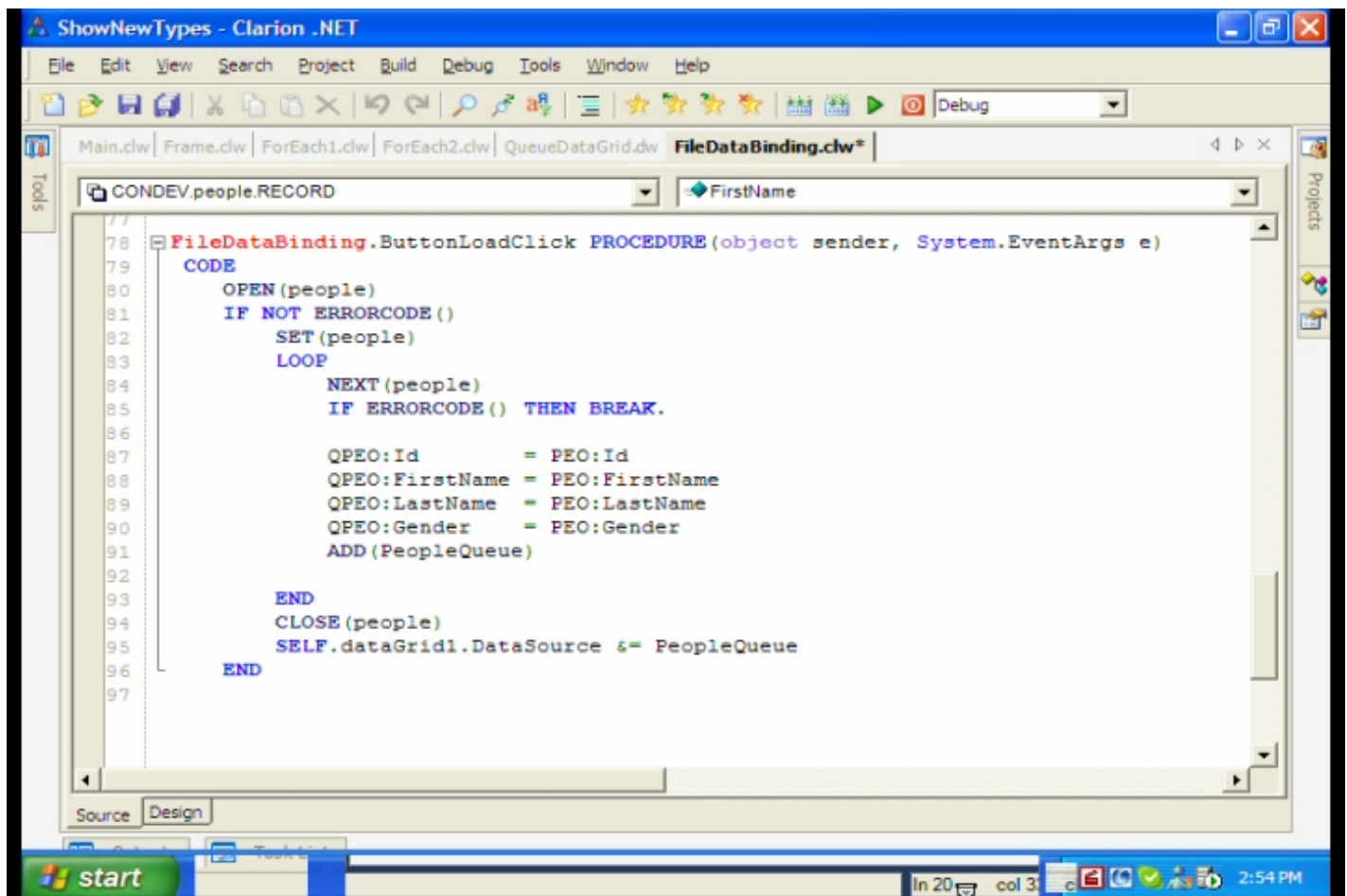
The running application with data grid. Since all of the WinForms components are available, you have a lot of choices as to how you want to present your data.

```

1  USING ('System')
2  USING ('CONDEV')
3
4
5
6
7  FileDataBinding CLASS (System.Windows.Forms.Form), TYPE
8  buttonLoad $System.Windows.Forms.Button, PRIVATE
9  dataGrid1 $System.Windows.Forms.DataGrid, PRIVATE
10 ButtonLoadClick PROCEDURE (object sender, System.EventArgs e), PUBLIC
11 CONSTRUCT PROCEDURE (), PUBLIC
12 InitializeComponent PROCEDURE (), PRIVATE
13 END
14
15 people FILE, DRIVER ('TOPSPEED'), PRE (PEO), CREATE, THREAD
16 KeyId KEY (PEO:Id), NOCASE, OPT, PRIMARY
17 KeyLastName KEY (PEO:LastName), DUP, NOCASE
18 Record RECORD
19 Id Int32
20 FirstName CLASTRING (30)
21 LastName CLASTRING (30)
22 Gender CLASTRING (1)
23 END
24 END
25 PeopleQueue QUEUE, PRE (QPEO)

```

Here's an example of data binding. with a class and a TopSpeed file. The TPS file data types are a wee bit different - CLASTRINGS instead of STRINGS, and an Int 32 instead of a LONG.



Again, the data is loaded into a queue and bound to the data grid.

My impressions

Although it's hard to get a complete picture until AppGen gets integrated, certainly the look and feel is much more like other contemporary IDEs. And I think licensing and modifying an existing IDE is a huge boon to SoftVelocity and Clarion developers. It frees up developer time to work on what SoftVelocity does best, and it puts Clarion on a level playing field with other IDEs. Based on the demonstrations, a hand-coder's version of Clarion.NET seems workable now, and I hope SoftVelocity releases such a product. Clarion 7 looks like the next logical step after that, followed by a full Clarion.NET AppGen release - given the changes between the Clarion and Clarion.NET languages, it's obvious that the templates will need some significant modification to begin taking advantage of the many opportunities offered by .NET and WinForms, so I wouldn't expect a full Clarion.NET AppGen release for some time. But don't forget that there were developers releasing hand-coded Clarion for Windows applications while that product was still in beta...

All in all, lots of good stuff here. And bring on that Clarion.NET hand-coder's release!

[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 produces the [Planet Clarion](#) podcast, which he co-hosts with Andrew Guidroz II.

Using Agile Programming Techniques for the Enterprise Information System : A Case Study

Louis Coraggio – Troy State University, Florida & Western Region
Wayne A. Lundeberg – University of Phoenix, Southern Arizona Campus

ABSTRACT

The authors describe the development of an Enterprise Information System (EIS) for an ISO 9001 manufacturing firm. The system is built using rapid application development tools with the method known as extreme programming. An overview of the EIS development process, the system design goals, and a chronological narrative of EIS development are presented. Included are additional requirements and recommendations for those considering agile methods.

AGILE PROGRAMMING CONCEPTS

Information management has a cyclical history that resembles the fashion cycle of men's ties; hold on to a tie long enough and it comes back in style. In the 1960s mainframe era, centralized processing ruled. Controlling access to resources and code efficiency were key IS goals. The introduction of the personal computer ushered in the distributed processing age. Users customized the environment and hardware became relatively cheap. Now that server based peer networks dominate the firm, centralized principles are being revisited as data integrity and controlling access are once again a priority.

In early application development focus was on getting the programming done, then worrying about documentation. The introduction of the Systems Development Life Cycle (SDLC) model and Systems Analysis and Design (SAD) methods focused attention on assessing user needs and constructing performance specifications prior to coding. The introduction of CASE development tools and object oriented programming techniques drastically reduced the expense and time for programming. The SAD methods are capital intensive, requiring a substantial investment of resources before producing a usable piece of code.

In the late 1990s, a movement toward "agile" software development gained popular momentum. The principles of the agile development are:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation; and
- Responding to change over following a plan. (Beck, 3)

Agile methods assume a constantly changing business and user environment. Focus is on cooperative efforts between programmers and the user community. Economic justification is the immediate benefit of the application, i.e. savings by the users will offset any increased cost for later maintenance.

Extreme Programming

The most widespread form of agile concept is Extreme Programming (XP) was first proposed by Beck (2). The formalizing of XP principles arose from the development of the

Chrysler Comprehensive Compensation system headed by Beck in 1997. Paulk (6) best summarizes the principles of XP:

1. *Planning the game* -- quickly determine the scope of the next release, combining business priorities and technical estimates.
2. *Small releases* -- put a simple system into production quickly. Release new versions on a very short cycle.
3. *Metaphor* -- guides all development with a simple, shared story of how the whole system works.
4. *Simple design* -- designed as simply as possible at any given moment.
5. *Testing* -- continually write unit tests that must run flawlessly; customers write tests to demonstrate functions are finished. "Test then code" means a failed test case is an entry criterion for writing code.
6. *Refactoring* -- restructure the system without changing behavior to remove duplication, improve communication, simplify, or add flexibility.
7. *Pair programming* -- all production code written by two programmers at one machine.
8. *Collective ownership* -- anyone can improve any code anywhere in the system at any time.
9. *Continuous integration* -- integrate and build the system frequently, every time a task is finished. Continual regression testing means no regressions in functionality as a result of changed requirements.
10. *40-hour week* -- work no more than 40 hours per week as a rule; never work overtime two weeks in a row.
11. *On-site customer* -- real, live user on the team full-time to answer questions.
12. *Coding standards* -- rules emphasizing communication throughout the code.

While neither of the authors had formally embraced the XP creed, both had an extensive history of collaborating on development of commercial applications. The authors had independently concluded that many of these principles worked. Pair programming, collective ownership, simple design, continuous integration, refactoring and coding standards had proven effective in past projects. With an on-site customer, much of the XP model was in place. The remainder of this paper describes the development of an Enterprise Information System (EIS) using (and discarding some) XP principles.

EIS DESIGN GOALS

Background

Catalina Tool & Mold (CTM) is an ISO 9001 manufacturer of precision plastic injection molds and molded plastic products. Historically, CTM has competed primarily on a quality and delivery basis. CTM is an industry leader in manufacturing technology, rapid delivery, and product design. The emergence of cheaper mold shops in the Far East has eroded the domestic market for traditionally price-sensitive customers. As a consequence, CTM focused on those customers for whom speedy delivery and/or extreme precision are primary concerns.

With annual sales of \$5-\$9 million and 60-90 employees, CTM was one of the larger independent tool and mold shops in the U.S. The typical job is a “one of”, make to order contract, done on a fixed price basis. On average, a job is in the shop for 6 weeks with \$150,000 in revenue. Repairs and modifications to prior jobs also constitute a substantial portion of sales.

In 1998, Mr. Lundeberg took over as CEO. Typical products at the time were cell phones, integrated circuit assembly trays, and industrial sprinklers. Customers pushed for faster turnaround times, experimental plastics, and closer tolerances. Investments in robotics, computer controlled machining and sophisticated CAD software could no longer be managed with the existing batch mode, report driven information system. The existing system was a patchwork of a Unix based, mainframe managerial/ financial accounting system, Excel spreadsheets, and user developed Access applications.

Much of Beck’s model fit circumstances at CTM. The network platform was clearly moving to Microsoft Windows NT based servers. CAD applications that had once required specialized UNIX workstations could now be run on Intel based Windows machines. It was clear that CTM needed an information system that would allow for enterprise resource planning and paperless manufacturing using real-time data. Strategic plans also called for re-inventing the business into new markets and products.

Objectives for the EIS

Extensive discussions took place with key operation people, the Board of Directors, and financial managers. After spirited debate, the following general goals were identified.

Job Estimation – Many design and manufacturing jobs lost money due to misquoting the project at its inception. A quick, clean, collaborative method for quoting new work was identified as the top priority. The new estimating tools should incorporate lessons learned from prior mistakes.

Real Time Resource Monitoring and Allocation – In general, labor and three key processes were identified as bottlenecks. Individual operations often vary greatly from budgeted time. This was true for both engineering and manufacturing operations. Additionally, management priorities changed based on new work; customer modifications; or errors in design and machining. Managers needed real time monitoring of shop floor resources.

Preparation for ISO 9001-2000 Standards – To maintain certification, ISO requires that: “(t)he organization shall plan and implement the monitoring, measurement, analysis and improvement processes needed to demonstrate: ... conformity of the product; ...; the conformity of the quality management system; ... and the continuous improvement of the quality management system.” (ASQ,1) The new EIS should include monitoring and exception reporting for non-conformance as well as archiving capabilities.

Universal Visibility – All workstations should have at least read privileges to most major aspects of the job.

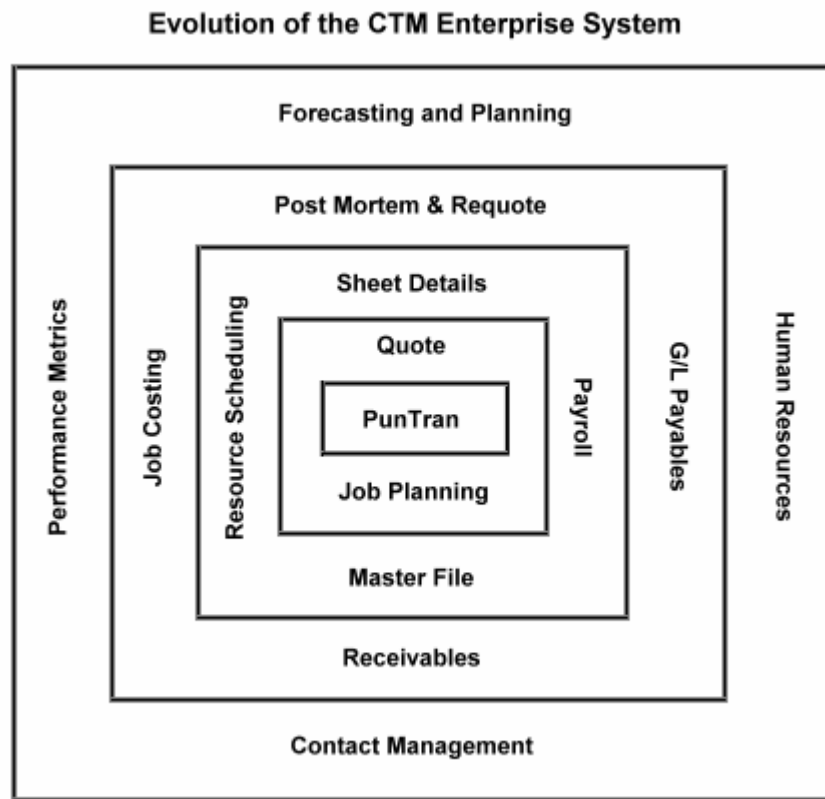
Cultural Compatibility – CTM had established cost centers, local terminology, and shop floor reporting formats. User acceptance and minimized training time were deemed mission critical.

Reduction of Redundant Entry – Much of the administrative information flow was paper to electronic to paper (e.g. time reporting.) Eliminating the initial paper step would reduce transcription errors and streamline accounting processes.

Post Production Analysis – Retrospectives on “what we did” to improve quoting skills, elevate quality and for use in performance evaluation. Analysis of the “as built” project would form the basis of a modular library of operations to be used in future job estimating.

Work began on the EIS in the summer of 1998. (Figure 1 depicts an Overview of the EIS development sequence.)

Figure 1



Development Environment

The Clarion environment is a 4th generation, object oriented (OO) development environment. Applications are based on the initial construction of a relational database with file relationships, formatting, and changeable referential integrity constraints. Basic forms, queries, tables, and reports are then constructed from the database design – much like Microsoft Access. Unlike Access, Clarion allows the developer to mix and match multiple file formats in the same

application and produce a stand-alone executable application that manipulates an independent set of data files. Also included is an independent report writer application that allows users to construct (and save) ad-hoc queries as formatted reports. Reports added in this manner do not require modifications of the database or recompile and reissue of applications. With a mixture of Windows 95, 98 and NT machines, Clarion's independent data and application provided maximum flexibility.

The interchangeable file formats allowed us to prototype quickly and rapidly convert live data. The proprietary "TopSpeed" file format was chosen for the final database. This format stores each table and all of its keys in a single DOS file and uses the application as a DBMS. Record hold/lock semantics, sharing, filtering and referential integrity constraints are enforced at the application level. Using this format allowed us to add, format, and manipulate individual tables without revising the entire database. It also places responsibility for maintaining database integrity and validity squarely on the developers.

The OO structure of Clarion provides an inherent consistency of formats, screens and reports. Much of the "code standard" advocated by XP is easily accomplished by customizing the templates used as a starting point for forms and tables.

Project Management for the EIS

In Clarion, the "Dictionary" construction is done independently from the "Application" development. When a Dictionary is changed, the programmer must close it, reopen the Application, and instruct the environment to integrate the changes. The Application must then be recompiled to produce the standalone "Program". Since the database is independent, each change of the Dictionary requires a conversion of the live database. With two developers, version control (and data file conversion) becomes a critical issue. To mitigate this issue, the authors created a "Master" zip disk. Only the programmer with the Master disk was allowed to change code and database fields. The official version was contained only on the master disk. Despite the availability of remote network connections, this master disk approach resulted in better personal time management and forced the developers to review and agree on changes before going live.

Each day, live data was backed up from the server to a development workstation. The live data and current versions of all Dictionaries, Applications and Programs were then archived to offline media. Each version revision was placed on a separate disk to allow the opportunity to roll back to a prior version if necessary.

While both developers worked on most aspects of the EIS, a comfortable division of labor was soon reached. As the "resident" member, Lundeberg was primarily responsible for defining business rules, final formatting of screens and reports, setting priorities, and developing program flow. Lundeberg had exclusive control of posting new releases and acted as project manager. As the "off-site" developer, Coraggio did most of the hard coding and prototyping. Database design and relationships were done together. The major benefits of paired programming were seen in developing complex database relationships and logically intensive batch updates.

User requests for changes and features rapidly got out of hand. Eventually, a formal request procedure was established to field requests. Each was reviewed by Lundeberg and typically forwarded to Coraggio for implementation in the next release. As the system grew, releases became less frequent. Many of the requested features were aesthetic or the result of disparities in local workstations. Since Lundeberg was both Project Manager and CEO, he had both the technical background and the authority to implement (or deny) changes in CTM EIS.

Phase 1 – Time Reporting

Customer pressure to reduce delivery times and price suggested the immediate need was an effective means of managing jobs in the factory. The existing Shop floor job management used a mainframe system driven by paper records of worker activity. Timecards served as both a payroll vehicle and as the primary record of cost center activity and progress. Workers typically put off filling out cards until days end, often with less than perfect recall. Transcription time added another 3-6 hours to the reporting process. As a consequence, a computer based timecard system (PunTran) was chosen as the initial system module.

The PunTran application drives most of the shop floor management. It was also the first time many shop floor workers would be asked to use PC's for cost accounting purposes. PunTran needed to be easy to use and relatively bullet proof. Workers punch in at the beginning of each shift. When punching out, net working time is calculated. Before a time record is accepted, the worker must completely allocate their time to current jobs and to cost centers. The construction of the cost center codes allowed the punch records to generate job utilization for key machine centers as well. A worker could punch in and out as often as they wished; eliminating the need to remember what he had done.

Introduction of PunTran was done formally in a company meeting. A general outline of the EIS was presented, and the initial PunTran release was demonstrated on 10 computers spread throughout the factory. A one-week period of testing was initiated. Workers entered time on both paper cards and in PunTran. At the end of the test period, worker comments were solicited.

Most found the system easy to use but slow. The prototype used list boxes and pull downs for validating choices. Most thought typing (with validation) should be an option. Workers also wanted to see a listing of their hours; to request personal time off; and to leave notes for management. All changes were implemented and the system went "live."

After a couple of weeks, behavioral issues came to light. Workers typically used one machine for punching in and another for punching out at days end. Word spread quickly that some PCs had slow clocks and others had fast clocks. In the morning, they used slow machines; in the afternoon they used the fast PC's resulting in as much as 15 minutes per day of extra time. (This was remedied by synchronizing local stations with the server clock)

Though individual workers had no edit privileges, several key administrators could fix "mistakes". It became apparent that someone was padding the records. Several security

measures were considered. Eventually, transaction logging was added for any edit changes and included fields that identified the date, time, workstation, and original values. Armed with transaction record logs, the culprit was identified and confronted. Word spread that the system had “spy-ware.” After about a month, most of the initial gamesmanship and “testing” was done. PunTran became part of daily life at CTM. PunTran freed one full-time clerical worker from time card processing.

Phase 2 Estimating and Job Planning

The job-monitoring database structure is the heart of this EIS. All resource scheduling, cost accounting, ISO documentation, resource utilization, engineering designs and purchasing are tracked and allocated to the job. Determining the granularity of the job database consumed many hours of analyst time.

Integrating estimating (“quotes”) with work orders (“jobs”) needed to be as seamless as possible. The Access based estimating system (already in use) was accepted and effective. Using it as a basis, the basic screen formats, program flow and logic were duplicated in Clarion. Using the Access structure also allowed us to convert live data into the TopSpeed format. This was particularly important because of the value of using historical quote information to aid in quoting similar new projects.

Both quotes and jobs may have several “items”, each having its own budget. (*Typical quote items include tasks like product engineering, mold design, mold building and plastic part production.*) Quote items formed the basis for job items, but a job may be assembled from a variety of quotes to the same customer.

Frequently, a job requires a many:many relationship with the customer file. CTM may ship to several locations around the world, and have several different contacts for billing, project management, or technical decisions. Some jobs (like repairs or enhancements) may actually be completed before written authorization arrives. As part of job planning system, contacts within a company were spawned into a child file. Also added was a communication log that attached to each contact and to each job. This allowed project managers to track the history of engineering changes and project anomalies.

Each job consists of a parent job record and two sets of children; the job items and one record for each cost center. When a new job is created, the basic information is copied from various quotes and transferred into job items. At the same time, one child record for each cost center is created. In the quote system, individual cost centers are carried as fields in the quote record. Thus each quote field instantiates one child record in the job. Multiple items are aggregated to their individual cost center. (Appendix 1 shows a partial list of CTM Cost Centers)

For each cost center, four fields representing the “hours” or dollars are included. There are five separate “hours” kept for each cost center activity.

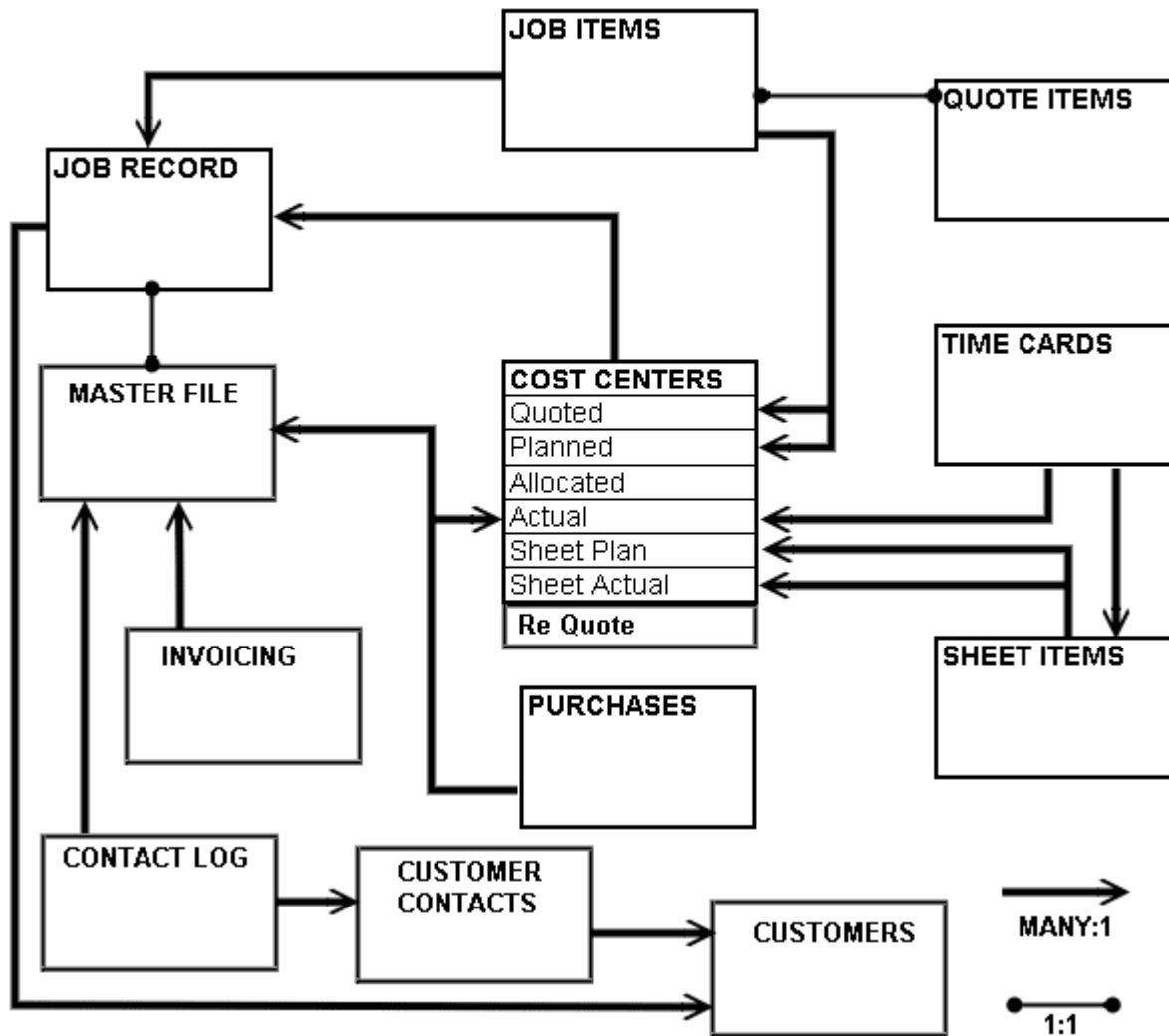
- Quote – Hours directly from the quote;
- Planned – Revisions based on conditions at the time the job is created;
- Allocated – Budget hours to be publicly visible (may be revised);

- Actual – Hours tracked by the punch system; and
- Requote – Analysis of the actual time spent per process restated to account for specific process events occurring on that job. The value of this is for quoting a repeat job.

Fields for start/end dates, remaining hours and management-only “fudge” fields round out the cost center record. (Figure 2 depicts the file relationships for the primary job files)

Figure 2

EIS JOB FILE RELATIONSHIPS



Phase 3 – ISO 9001 Subsystem and Documentation

The completion of Phase 3 stabilized the database and established the central transform for the EIS. It allowed the developers to divide work independently as well. ISO 9001 compliance standards require considerable documentation of meetings, reviews, customer contacts and error correction processes. Most of these are tracked to the job level. In a

normalized database, most of the ISO requirements should be part of the main job record as most requirements are 1:1. However, the sheer number of fields, and their infrequent access, led us to create a separate “master file” record for each job. This record also became the parent for most of the administrative tasks including purchasing, shipping, inventory movement and customer invoicing.

From a culture standpoint, most of the ISO documentation is after the fact. Consequently it becomes a nuisance. Much of the development work was targeted to making the job tracking record a push button operation. The CTM ISO System calls for a series of reviews and meetings throughout the project. When the master file record is created, tentative dates and participants are instantiated for these reviews. Based on the ISO plan and job characteristics, appropriate fields are displayed or hidden from users. If the program detects that a target date has past, users are prompted for an explanation or asked to fill in details.

Splitting the production and administrative sides into separate records also enhanced network performance. Typically, managers worked on the job tracking side while manufacturing floor people worked in the cost center side.

Phase 4 -- Resource Scheduling

Rolling up all budgeted and actual hours to cost centers provides a good overview of individual jobs and resource loading. However, it does not provide sufficient detail for scheduling. Since short delivery time is a major competitive advantage for CTM, scheduling is the mission critical function of the EIS. In manufacturing, tasks are identified on individual sheets of the design drawings. A prototype mold may have only one cavity; a production mold may have several cavities. The estimates, specifications and CNC programming for a single cavity might constitute a “Sheet”. For a six- cavity mold, each sheet detail would be repeated six times. Thus the sheet detail becomes the atomic unit for purposes of resource scheduling.

In a perfect world, sheet details would roll up to job items. Job items would roll up to the main job record. CTM history had proven otherwise. Competition for resources, customer changes, and rework all create a dynamically changing manufacturing floor. So another set of child records were created to accommodate shop floor scheduling. Sheet detail records are logically the children of the main job record. As cross check, sheet hours are also rolled up to their respective cost center records.

With the complex relationships and evolutionary development cycle, job records are not truly real-time. Job records are actually batch updated on-demand at the user level. Each time a user asks for details on a job, a batch process checks for changes and additions in sheet and timecard files, updates the job record in the database, then the record is displayed for that user. Since Clarion does all database management at the application level, this approach offers improved performance for all users, without forcing every entry in the PunTran system to update multiple tables.

Several versions of the resource scheduling process were modeled. Scheduling simulations were tried (based on PERT and Shortest Processing Time). While theoretically

attractive, these proved impractical due to uncertainty of individual workstation times; on-the-fly redesign of the product (concurrent engineering); and resource availability. These prototypes were built with sample data, and demonstrated in focus groups with line level managers. This departure from “simple design” principles proved to be a major waste of time. This was a definite win for the XP camp.

The final version uses a list based priority system that employs a group technology approach. Resources (labor and machinery) were classified into groups based on interchangeability. Cost centers were first aggregated into groups based on similarity of skills and workstation capabilities. Each resource was assigned to one or more groups based on capabilities. For each job, tasks are also assigned to the same group designators. This allows managers to look at subsets of resources and tasks that match up. The assignment is by drag and drop. The sequencing of tasks is determined on a daily basis by the plant managers through assignment of a priority.

Phase 5 - Financial Accounting

For the financial accounting functions, a set of Clarion “templates” was purchased from another developer. These included General Ledger, Payables, Receivables, Payroll, and Purchase Orders. The authors spent about 150 hours customizing these templates. Much of the time was devoted to figuring out the template logic and data structures used by the original programmer. Importing time card records for payroll and invoicing for receivables are the primary interface with the main job based system. The authors found dealing with “foreign” code frustrating at times, the time and manpower savings from the purchase of these templates clearly justified the decision.

Payroll was the first subsystem brought on line. With PunTran in use, all of the pertinent information could be electronically accessed. Once the payroll system was customized for local taxes, direct deposits and payroll deductions, a batch process extracted payroll hours from the PunTran database to produce paychecks.

Receivables are handled within the Master File records. CTM agreements typically contain progress payments, and recurring invoices that required tie-ins to job items, quote items, and authorizations already contained within the main job system. Only summary information is exported to the General Ledger. Payables are handled within the financial system by the purchased template system. Balance Sheets, Income Statements and other standard financial reports are maintained in standard spreadsheet programs.

CTM operates using a revolving line of credit secured by capital assets and receivables. Periodic reviews of key financial ratios and performance indicators are part of the bank covenants that dictate interest rates and fees. Since CTM has a mix of recurring production and one-time projects, determining revenues, costs and work in process is a quite complex. Accurate measures of true output are performed on a monthly basis. Sales taken for a month are determined on a job-by-job basis by management. Once sales are determined, an extensive batch process rolls up all payables, invoices, time records and job progress. From this Shipment/Work In Process (WIP) report, financial ratios are calculated and the labor overhead burden is

determined. Each quarter the shop labor rates (containing the overhead burden) and machine rates are reviewed and adjusted based on Ship/WIP results. The completion of the Ship/WIP process was the final link in the enterprise system. It ties the managerial accounting system to the financial accounting system and generates most of the performance metrics required by management.

Phase 6 - Marketing and Performance Evaluation

As the EIS evolved and manufacturing floor management became more streamlined, the primary focus turned to marketing. By mid 2001, the CTM client base had shifted. Jobs were smaller, with a wider customer base. CTM had opened a subsidiary operation in Queretaro, Mexico to service major manufacturers in the area. CTM marketing efforts turned toward soliciting ideas for new products from patent holders and high margin specialty manufacturers. This became known as Concept To Market (C2M).

Contact management, follow-ups, forecasting and long term monitoring of market segments became critical to the success of the C2M strategy. For the EIS, a contact management scheme was added to track personal responsibilities, dates, and prospect qualification information. Likelihood estimators for quoted jobs were added to improve sales forecasting. Much of the development effort was directed to making contact system easy to use and convenient. Reminders, and contact information buttons were tied to customer files, quotes, job progress, Rolodex and calendar functions. Wherever possible, field values were instantiated based on login, program location, and customer information. Exception reports and color-coded browse lists enable senior management to assign responsibilities and set priorities for both the corporate and individual to-do lists.

Postmortem job analysis and ISO documentation include both “as built” information and mitigation for any quality issues encountered during construction. This provided an opportunity to add individual performance appraisals for employees. These became the basis for the HR sub system. In combination with time summaries from PunTran, wage/benefits from payroll, and personal information, these became the Human Resources system. The addition of user-defined queries enables management to look at histories for individuals or groups, as well as a cross-section for any time period.

Security

While most of the EIS information is visible to all employees, there are areas limited to senior management, and accounting. Three methods are used to secure access to sensitive information, passwords; separate applications, and hidden menus. A master “switchbox” provides access to all of the individual applications. When the user logs into the switchbox with an ID and password, only those menu choices available to that user are displayed. The switchbox in turn calls individual applications and logs the date, time, station, and user in a transaction file. Individual applications also look for specific “flag files” in other folder locations to verify that they are being accessed from the CTM server. Absence of these flag files will cause an immediate shutdown.

Within an application, there are selected menus that only appear when the user executes a specific key sequence. This method is primarily used for senior management to “adjust” budgets, editing timecards, or printing sensitive reports.

The structure of the TopSpeed data files also allows the database to be copied while in use. Thus mirrored copies can be quickly generated for tracking and comparison purposes. The EIS resides entirely on a single server, with firewall protection to the outside world provided by a Microsoft Proxy Server. Source code is kept on a single local machine. While a knowledgeable employee could conceivably make copies of the data and applications, it’s unlikely that he/she could reproduce the necessary local flag files required for operation.

Refactoring Examples

As more and more of the daily activities at CTM were included in the EIS, user performance expectations also rose. Increased dependency on the EIS created higher demands on the applications. Requests for unusual reports, local use features, faster data input and different layouts dominated requests. Many of these requests were not judged to be worth the developers’ time, or implemented and never used. Though Clarion includes a “Report Writer” for ad-hoc queries, instructing users in its operation (and the structure of the database), proved impractical.

For recurring views, like receivables, export queries by example are employed. Users can tag individual fields in tables, and export the results to comma-delimited, ASCII text files. Resulting files are then imported to Excel or Access for further processing. For “one time” queries (e.g. a capital loan application), the developers created small programs to extract the data and process it with standard office software.

As the EIS grew, infrequently used reports and queries were moved into a separate application to facilitate code maintenance, and enhance performance and response-time of the core procedures. These procedures characteristically require massive sequential searches of tables, intense screen I/O, or creation of temporary files. Though this module can be called from other applications via embedded command line, it also acts as a standalone application. Thus long searches (like the Ship/WIP report) can be accomplished as a background task. Unused procedures and one-time transitional code was removed and archived.

Final Form of the EIS

When the “final” version of the EIS was released in summer of 2002 perhaps half of the U.S. mold building capacity had disappeared. CTM had survived the exodus, and won ISO 9001-2000 certification – one of the first mold shops in the world to do so. Sales for custom machining, OEM products, and C2M projects have supplemented traditional mold building.

Sales for 2001 were only 85% of 1998 levels. Over the same period, revenues per direct labor hour rose from \$53.16 to \$63.62. Much of this rise in efficiency can be attributed to the EIS. Figure 3 shows a data flow diagram of the final system. In Figure 4, summary statistics on development are provided.

Figure 3 Final EIS System Overview

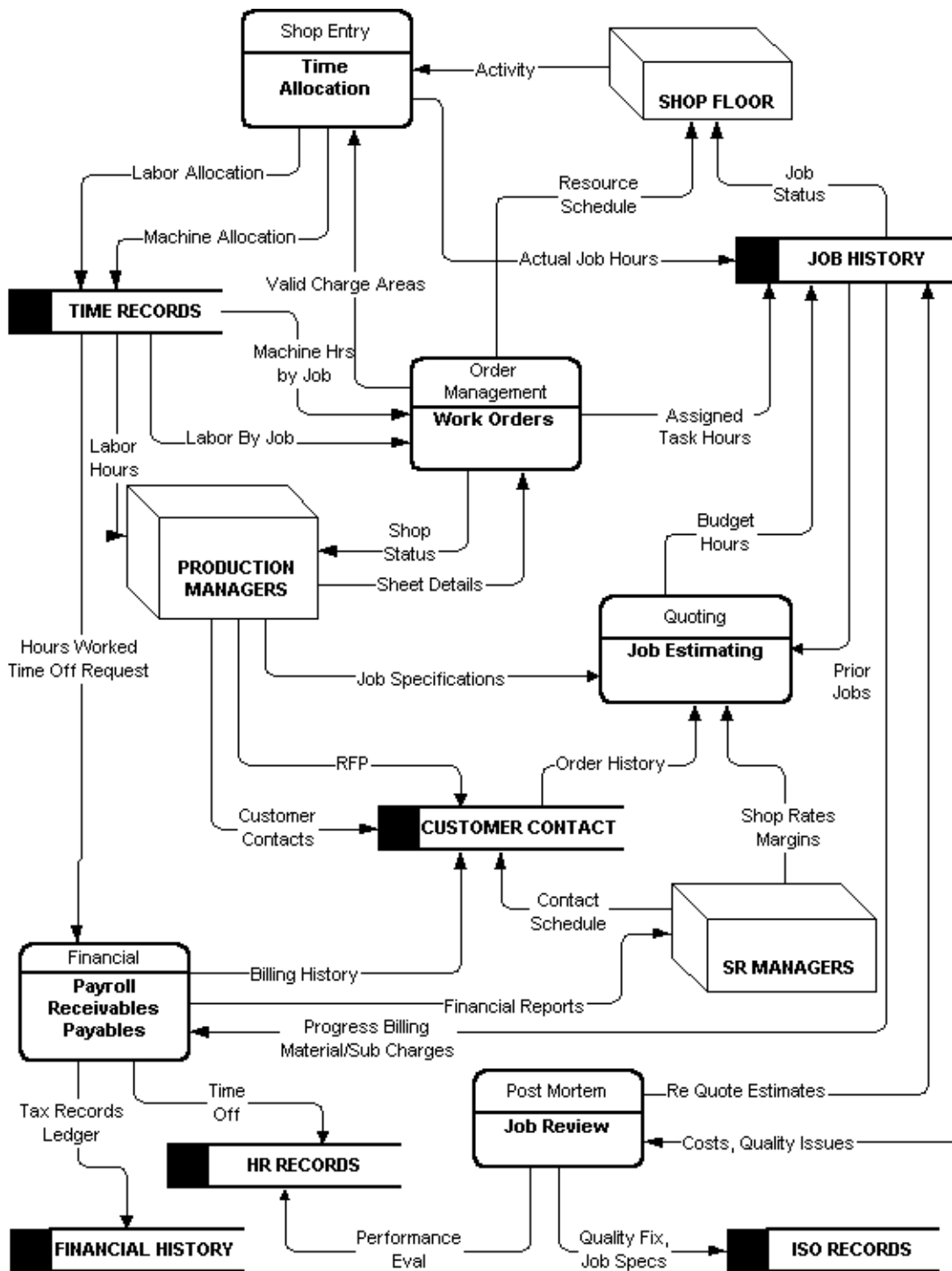


Figure 4
CTM EIS Final Statistics (June 2002)

Procedures in Financial System	275
Procedures in Main System	303
Primary Tables in Data Base	150
Externally Generated Reports	94
Estimated Programmer Hours	1400
Approximate Development Time	22 months

SO DOES XP REALLY WORK?

The CTM EIS must be considered a major success. Since the EIS development follows the XP model, the authors' answer would be a qualified yes. The biggest flaws in XP reasoning are the notions of the simple metaphor and simple coding. Stevens (7) makes a strong case for lack of planning being the major downfall in XP projects. The authors concur. Even if no formal documentation is created, time spent on the data dictionary, entity relationships and data flows are necessary planning stages. Most of the rework time was consumed in revisiting older code to add new functionality. Beck suggests that system specifications be assembled from "note cards" containing user requests. While that may be a good starting point, a trained analyst must convert those requests to fields, records and processes.

The simple coding concept ("exactly what is needed now") should be a guideline, not a rule. Adding database fields for future use, and hooks for future processing are usually viewed as major time savers. In XP style development they also serve as part of the development plan and reminders for procedures yet to be done. The inclusion of "to do" procedures also acts as a preview of coming attractions for the user community.

Paired programming has definite advantages for database development and complex logical modeling. Cockburn and Williams (4) suggest that the cost of a second programmer adds about 15% to the cost of development, offset by a 15% improvement in quality. Collaboration also ensures consistency in coding and data element naming conventions. Once the central transformation is functional (job planning), much of the development focuses on additional input and formatting output. Report writing and screen design are more modular tasks and could be delegated to additional programmers. However, the logistics of meeting, communicating, and integrating the pieces could rapidly consume any extra capacity. Having both resident and non-resident members proved valuable. Intricate and lengthy procedures are better accomplished off-site, with no interruptions. A resident developer can more effectively do testing, revisions, formatting, and refinements.

Based on the experience of Catalina's EIS the authors suggest the following additional precepts for those considering an XP approach.

The development team needs decision-making authority. – To consider all of the user requests and potential policy changes, decisions must be made quickly and have force of law. For XP development, a benevolent dictator is more efficient than a democracy.

Analyst and programmer roles must be tightly integrated. – Many of the flow and logic decisions require an understanding of user perception, operations reality and programming feasibility. Adding new capabilities often creates new software performance specifications and changes in business rules. The analyst’s role as user liaison is vital to acceptance.

Anticipate transitional nulls. – As new tables, features and fields are added, the transition of older data must be incorporated into the code. The XP approach mandates checks for null data and some method for dealing with it. In most cases, this transitional code (or field instantiation) goes away over time. However, transition management will consume a substantial amount of coding time. The majority of incremental programming cost under XP (over SAD methods) occurs in this transition management.

Create a release plan and adhere to it. – As the system grows, incorporating changes and making the user community aware of those changes becomes a management task in itself. At CTM all requests for changes had to be submitted in writing. These were batched together and released about every two weeks. A log of changes was attached to the application, and the current release date displayed on the application title bar. New system features became part of production planning meetings.

A motivated user community is a must. – The technical sophistication of the workforce and the “can-do” corporate culture at CTM is largely responsible for the success of the EIS. Direct user requests and feedback were largely responsible for many of the EIS capabilities and refinements. Without an active, receptive user community XP development will flounder.

Save a rollback version. – Despite thorough testing, every new feature added does not prove successful. A complete sequential backup of applications and data will be a major time saver. It’s often easier to start over with a prior version, than to undo changes.

Be flexible in database design. – While the theoretical advantages of normalized database design are well documented, they may become restrictive in practice. The addition of 1:1 file relationships, extra fields, and multiple key relationships may result in major performance gains and greatly simplify development efforts.

Allow adequate time for organization adaptation. – The default setting for the user community is “I don’t know what I want, but I know that isn’t it.” Introduction of a new report or subsystem will inevitably meet resistance. During the CTM development, users were told that changes would take one to three weeks for completion. Many initial reactions disappeared after people lived with the new report for a while. The lag between activation and utilization will be longer when there is a culture change involved. Though considerable effort went into simplifying the contact management subsystem, many users simply don’t use it.

Hold on to back up systems as long as possible. – Some bugs may take weeks or months to surface. Issues with triggered events (like year end closings or quarterly taxes) may reveal problems that were not caught during routine operations or pre release testing. Trial closings with backup data are highly recommended.

Consider batch, on-demand processing. – Many reports are infrequently needed, or are limited to very few users. Using spawned, batch-style processes, (with locally generated files), for these reports will save extensive rework of database keys and relationships. Batch processes are typically easier to modify for transitional nulls.

Limit end user computing. – While major development is going on, users will insist that provided reports and screens don't meet their needs. Some will create shadow systems without recognizing changes in business rules and the effects of transitional nulls on data integrity. If the need for ad hoc figures arises, the development team should provide it. Agile development will exacerbate reconciliation problems.

Consider user documentation issues. – Most programmers and analysts despise doing user documentation. The accelerated pace of XP methods exaggerates this problem. At CTM several key users act as testers and become “Local Resident Experts” (LRE.) While formal training was regularly scheduled, the LREs provided the majority of instruction. In commercial, “shrink wrap” situations, this approach may not be feasible.

References

1. American Society for Quality, *American National Standard Quality Management Systems – Requirements for ISO 9001-2000*, December 13, 2000, Quality Press, Milwaukee, WI
2. Beck, K., *Extreme Programming Explained: Embrace Change*, Addison-Wesley, Reading, MA, 1999
3. Beck, K. et.al., *Manifesto for Agile Software Development*, <http://www.agilemanifesto.org/>, Confirmed October, 2002.
4. Cockburn, A., Williams, L., *The Costs and Benefits of Pair Programming*, XP2002, <http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF>
5. Hoffer, J.A., George, J.F., Valacich, J.S., *Modern Systems Analysis and Design, 3rd Edition*, Prentice Hall, Reading, 2001.
6. Paulk, M.C., *Extreme Programming from a CMM Perspective*, IEEE Software, Vol. 18, No. 6, November/December 2001, pp. 19-26.
7. Stephens, M., *The Case Against Extreme Programming*, http://www.softwarereality.com/lifecycle/xp/case_against_xp.jsp, Feb 3, 2002

“Clarion”, and “TopSpeed” are registered trademarks of SoftVelocity Inc.

“Access”, and “Windows” are registered trademarks of MicroSoft Corporation.

Appendix 1 -- PARTIAL LIST OF CTM COST CENTERS

General Labor		Machine Centers		Materials	
1	Layout	141	CNC Electrodes	201	Mold Base
2	Detail	150	EDM Sinker	202	Steel/MB Plates
3	Checking	151	Mold Base CNC	203	Pins/Comp
4	Electrode Design	152	CNC Machining	204	Graph/Wire
5	Programming	157	Mold Sample	205	Hot Runner Sys
6	Training	158	Production Molding	206	Misc
7	Run Prints	160	Wire Steel	207	Tools
8	Database Processing	165	Wire Electrodes	208	Resin
11	CNC MB				
12	Conventional MB				
21	CNC Machining				
22	Conventional Machining				
23	Grinding				
			Administrative		Subcontracting
41	CNC Electrodes	506	Inspection Admin	301	Heat Treat
42	Conventional Electrodes	510	Engineer Admin	302	Thr. OD Grind
50	EDM Sinker	511	Indirect	303	Turning
60	Wire Steel	512	Supervision	304	Jig Grinding
65	Wire Electrodes	514	Meeting/Training	305	Plating
70	Honing	517	Shop PLT	306	Gundrilling
75	Heat Treating	527	Project Management	307	Polish
80	Polishing	618	Admin Salary	308	Mold Design
90	Assemble	622	Admin Holiday	309	Text Engrave
91	Meeting	623	Admin PLT	310	Sample
111	Production Molding	624	Admin PTO	311	Freight
112	Mold Sample	217	Personal Time Off	312	Misc Subcontract
120	Inspection	218	Shop Holiday	313	Wire - EDM
121	First Article Inspection			314	Machining

Clarion Magazine

Planet Clarion Transcript: Bob Zaunere Part 2

Published 2005-07-28

Editor's note: This is the second half of the June 2005 podcast. The first half is available in both MP3 and transcript form, but due to production issues this second half is *only* available in transcript form.

Dave Harms: Now, we started talking about all this when the subject of COM came up, and COM controls in the IDE of in live data and that sort of thing. In a .NET environment, I gather it's a lot easier to have live data in your... say in the window format or in the forms designer.

Bob Zaunere: That is, yeah, that's absolutely true.

Dave Harms: But do you have any plans to implement that kind of think in Clarion 7 or is it that a situation where the work involved would be pretty massive.

Bob Zaunere: We don't have definite plans to do so. You know, the... talking about live data as opposed to COM controls. I haven't seen a whole lot of demand from people wanting to see live data. That wouldn't be, I don't believe that would be very difficult to do by the way. Fairly straight forward. As far as the COM control support, I don't know. It's a question of demand and time. People want things, you know, yesterday, and, yeah, there's certainly some demand to say oh, I wish I could drop COM controls into my WIN32 code. But I wonder how that will change with Clarion.NET is shipping, and how many people would rather have... move to that forward into .NET.

Dave Harms: Yeah, this is the thing, and this is my personal bias which certainly is not shared with Clarion developers. When I look at... the thing that I liked the best about .NET is that to me it removes... I guess what I'm saying is I think is COM is perhaps the best argument for .NET, you know. When I look at...

Bob Zaubere: I'd have to agree with you 100 percent.

Dave Harms: When I look at how difficult it is to implement COM in any application, boy I just I think I'm really looking forward to the ease of use of Clarion.NET.

Bob Zaubere: I would agree with you, it is. And there's quite a few people who were former COM gurus, if you will, who enamored with how much better .NET is. Aside from the fact the implementation is much better done, but additionally because there's documentation. It's just a whole different implementation. Microsoft, when they did COM never even finished all the documentation. It's a pretty well known fact that they just didn't have enough staff on that particular project to devote to completing the documentation. That's why there's a proliferation of after-market books on COM because people had to figure it out and imply how it worked, and go from there. So I tend to, you know, I tend to think we'll take a wait and see attitude on how important getting COM into Clarion 7 is. You know, there's good improvements in Clarion 6 as far as that goes with the, in general, COM has worked well <inaudible>.

Andrew Guidroz: There are improvements in working, but you know, to me what most of us out there need is improvements in implementation to make it easier for us to drop these things in because, you know, I did the article for Clarion Magazine on just using a very simple CD burner, and it was a non-trivial task to...

Bob Zaubere: I do recall. We even sent you a little...

Andrew Guidroz: You had some documentation, and all sorts of great things that, you know, and I battled, and you all end up sending me example apps, and for me to do it and show it also on Jim Kane's thing. A lot of interaction with him.

Dave Harms: And you know how easy it is to do the same code in .NET, in Clarion.NET, would we have it? It's about a half dozen lines of code. You instantiate an object, and you call it an addFile method as needed, and then you call a write method, and that burns your CD. Now what's interesting about that is that under the hood, if you look at this assembly that you can get from MSDN to do that, that's it's actually a wrapper for the COM code. But the point being is that you've got these .NET classes whether they're fully native or whether they wrap some API or some COM stuff is that classes to do these things are readily available, and so much easier to use than the COM version.

Bob Zaubere: One hundred percent true, and that applies across the board for existing

COM objects, but using any... if you look at .NET controls, their counterparts, in most cases people who have put it on COM controls had created .NET components that have equivalent functionality for use in .NET as a managed code. And if you look at how easy it is to use the counterpart, the matching component under .NET it's like night and day. So it's very clear, you know, you write some very clean and clear code. There isn't a lot of mystery to it. It's just much nicer.

Andrew Guidroz: So we can say that language developers have in fact learned a few things over the last decade.

Bob Zaunere: Absolutely, I would agree.

Dave Harms: Is C7 really going to be you trying to reinvent the wheel on a lot of this stuff or is it going to be more focused on a new IDE. I mean it's...

Bob Zaunere: Well, you know, C7 really is just more of a, you know, just continue to evolve and refine the product that we have today. There'll be some focus on the cosmetics, I think that obviously it's proven to be very important to people. Additional graphics type supported, additional options on for instance icons in there and how they look when they're disabled, enabled, and hot/cold that type of thing. But there won't be any radical changes, it'll just be, you know, there'll be improvements in templates, there'll be additions to the template area, but...

Dave Harms: And with the flattening of the IDE, we're talking about some pretty major improvements in productivity as far as developers go.

Bob Zaunere: I think we're talking about huge improvements in productivity. I know I saw at least one or two comments that someone kind of questioned, you know, well what will happen to our productivity. I guess I only took that as one more indication that, you know, without a context things can be misinterpreted because I really, you know, I agree with what you said. I think this will be the single biggest leap in productivity.

Dave Harms: Well, I was just talking to a developer yesterday who said, you know, to him that was the biggest issue with the flattening of the IDE, you know, how long would it take him to make back that money in time, and "not very long" was the answer. So, yeah, I think that's got to be one of the biggest features of C7. How about reporting? How does the reporting end of things look on Clarion.NET, or just on .NET in general?

Bob Zaunere: Well, without question built in reporting is... will be in Clarion.NET. It's

an important part of Clarion in my opinion. Something that isn't present in, I don't know, I don't know every language at that level, but I don't think you'll commonly find it. How it will end up implemented is... and what improvements we've made is still under investigation. It will definitely be implemented, but it will be when Clarion.NET ships will definitely be support for reports, report structures, at least today it looks like report structures. But there will be reports.

Dave Harms: Is there any kind of a standard way of doing reports under .NET, or is it kind of everybody rolls their own.

Bob Zaunere: I would say everybody rolls their own from what we know. But, you know, again, we have a, you know, we have a pretty good report engine. It's a very capable little beast. It certainly has room for improvement. Everything in the world does. But I guess the point to be made here is that Clarion developers can rest assured we'll have reports. More interesting to me would be what would you like to see changed? That would be of great interest.

Dave Harms: And then both from the end user and from the report writer interface point as well as features I guess eh?

Bob Zaunere: Yes, particularly the functionality. Certainly the designer part is guaranteed to change to some extent.

Dave Harms: Will there be an interface into the new IDE for batch compilers and that sort of thing, or will just the way things are done with solution files not really make that important anymore?

Bob Zaunere: You know, that's a good question. From my perspective, there's... it somewhat eliminates the need for batch compiles because you can have a solution that can contain as many projects as you can care to <inaudible> within the tree.

Andrew Guidroz: Hooray, hooray, Clarion 7 as well or just the .NET side?

Bob Zaunere: Clarion 7 as well for sure. I would have shown that. That was one of the niceties that didn't get shown, a compilation of a solution file with multiple projects under Wink 32. So that definitely will be there. The only other thing that's ever been brought up, it had some interest, I don't know that... I don't know how wide spread it is, but the idea of a distributed compile, i.e. being able to do compiles upon multiple machine, has, you know, it has kind of some interesting aspects to that. I don't know that it will be something

we'll drop into the first incarnation of the new IDE, but we're definitely considering that as something that, you know, maybe it's an interesting piece of functionality that we've kind of architected out how it looks like it could be done pretty reliably, and you know, that has more interest to me than the batch compiles.

Dave Harms: I guess you...

Bob Zaunere: By the way, the solution files, just one last point, I keep referring to projects, but I think of a .APP in my mind, often I just interchange them. So a solution file can include multiple APP files as well.

Andrew Guidroz: Okay, well that APP question brings something up that I asked you back at ETC, at least at ETC, maybe even before that, the APP file and the dictionary file, is that going to remain relatively the same, or you're thinking about doing some major changes there, like even leaning towards XML?

Bob Zaunere: Well definitely internally the dictionary file has already changed a great deal because there's new functionality added to it. The APP file will definitely change as well. I don't know that we're going to move it into XML. We might have an export to XML. The APP file is a very complex representation. The dictionary... it's not trivial by any means, but it isn't nearly as complex as an APP file. But some would have to convince me of what the benefit of an XML representation is as opposed to a TXA or a TXD representation. What would you say Andrew?

Dave Harms: Reporting by XSLT?

Andrew Guidroz: Yup. Just, there's some robustness that there's... it's a, you know, it's not uncommon to corrupt a dictionary or an APP file, and it's hard to recover.

Bob Zaunere: I agree with you in that, that it does happen, and it's disastrous when you don't have a back up.

Andrew Guidroz: Right. So literally when you start talking about XML or some other more common file formats, like even ZIP...

Dave Harms: If the problem is corruption of data, you know, the data somehow gets mixed up, the data can get mixed up no matter what you store it in, so I don't see how XML would...

Andrew Guidroz: Well the trick with XML is if I got 65K still sitting out there as opposed to my 68K if it was complete, I probably can recover 65K and do something with it, and just rebuilt, you know, the last 3K that I lost. And that's...

Dave Harms: You think?

Andrew Guidroz: Yeah, that's...

Bob Zaubere: Yeah, with a dictionary I'm certain you could do that. With an APP representation, might be, it might be a nontrivial task.

Dave Harms: Yeah. Seems me you still need good back-ups no matter what.

Andrew Guidroz: Well no, I understand the concept of back-ups, but I mean evidently, I mean I don't know if you're reading a lot about what's going on with Office 12 with Microsoft.

Bob Zaubere: I have not, no.

Andrew Guidroz: They've committed now that they're... every document, whether it's Word's documents or Excel spreadsheets or a PowerPoint presentation, they're committed 100 percent to go to number one XML, and number two, it won't be a pure text XML...

Dave Harms: Yeah, it's fascinating what they're doing.

Andrew Guidroz: They're going to take that text and zip it. So literally all your files that you're going to pass around will ZIP files of an XML structure.

Dave Harms: Yeah. And you can basically have sort of nested ZIPs inside that too, right. So what you have is you have one area, you might have an XML file that contains some, you know, some... well it might contain the document, or it may contain some other related information, and you could have several XML files, you have another directory in the ZIP for images, and if you want to... you have another area for macros. And another one for comments, notes and things like that...

Andrew Guidroz: It's a repository.

Dave Harms: And so, yeah, so it's a repository basically. And you can easily strip out parts of that if you want to say send a document to somebody without the comments.

Andrew Guidroz: I'm just, you know, Bob, is it critical? I don't know, I think that the easier it is for us to massage data makes it easier for say third party guys to get in the middle of this thing and do a little massaging.

Dave Harms: Of course it makes it easier for them to screw it up too, so...

Andrew Guidroz: It does. But also the data recovery aspects of it that, you know, from time to time I still read up there, and you know, I have an APP file and I can't open it anymore, what should I do, and everyone says go back to your back-ups, and sometimes you just wonder if man, if could... if I'd loose those last three procedures at the end of this file, it wouldn't break my heart, you know.

Bob Zaubere: Right, yeah. It's a topic that we've, you know, there are two areas. The recovery part of the comment, without question, we've already built into the dictionary and we'll be plugging the AppGen as well, the idea of an automatic back-up in intervals so that people will not have to think gee, did I save this yet? It's going to happen, and it won't just be, you know, the single back-up that we currently have because of course it does exist currently. So that will be there. So I think that will definitely account for a lot more safety in terms of data recovery. In terms of TXA versus TXD versus XML or some other representation, I think that that is probably something achievable for a dictionary. For the APP file it's a little bit more complex. If the idea was simply for robustness, I'd say we probably don't... it's probably not needed. If the idea is to work within the context of a text file, okay, there's some validity there, I'm just not sure how often people would really say, you know, I want to get that APP in XML so I can start working on it, I need to do something to do it, as opposed to recover it. So, you know...

Dave Harms: And it's not like there isn't a way to work on them now, you can always export to TXA and work on them and put them back in. What will they... you said we don't probably need an interface to the IDE for batch compiles, but what about, there would still be an interface I guess for registering templates and so forth?

Bob Zaubere: Yeah, the entire, of course templates will still be integral to the system, so yes, without question.

Dave Harms: Yeah, and will that be the same mechanism as you use now or will you use a different interface?

Bob Zaubere: I don't see any reason to change how that works, so it'll be the same.

Andrew Guidroz: So we got one other question in regards to the new IDE, and I think that's Data Modeler. I mean does that continue even in C7, the same Data Modeler or...

Bob Zaunere: It will not, no.

Andrew Guidroz: It will not. So you will just drop that part of the product?

Bob Zaunere: Absolutely. There's a lot of work being done on the dictionary that accounts for everything that was in Data Modeler and then some.

Dave Harms: Oh really.

Bob Zaunere: No, I don't think that anyone will miss it.

Dave Harms: So there will be visual design capability in the C7 dictionary?

Bob Zaunere: It won't go that far. There is a start on working at a visual level, but quite honestly, I haven't found that the visual representation in Data Modeler was the way I like to work with it too much. When the introduction to the subsets happened, that was a big step forward. But there will be some visual capability. Yeah, I won't go into detail on that, we'll save that for another podcast. But one thing I will say is that the working with dictionaries will change for the better a great deal. You can work with them as you do now, but they'll be some real productivity gains in how you can make changes and look for, and not just search even though in the searches in C6 are vastly improved over anything we had before in the dictionary. In Clarion 7, and of course that applies to Clarion.NET as well, they'll share that component. It will be so much nicer to work in the dictionary.

Dave Harms: Cool. Are there any things as far as the IDE, C7, or Clarion.NET, are there things that we haven't touched on that you wanted to mention?

Bob Zaunere: Well there's a lot of other components that no one has brought up, and I just as soon just save them for another time. But I'll leave you with this tidbit. There are components that no one's asked about that I don't know why they haven't. There are things that don't exist in SharpDevelop by the way or Clarion 7 that will exist in our final IDE that we can talk about at another time, but there's definitely going to be quite a few new components.

Dave Harms: There's a teaser.

Andrew Guidroz: Well, let's jump...

Dave Harms: Yeah, and now we come to the inevitable part of the interview where we ask about...

Andrew Guidroz: Dates and prices.

Dave Harms:...dates, the prices, and all that stuff. Do you have an anticipated date for C7 beta?

Bob Zaubere: I don't really... I don't want to say that I don't have an anticipated date, but you know I get in trouble every time I say anything about a date. So...

Andrew Guidroz: Only when if it doesn't happen on the date. <laughter>

Dave Harms: And of course, none of the rest of us ever miss our delivery dates, so that's...

Andrew Guidroz: You don't have a ballpark?

Bob Zaubere: What I think will happen at this end of it is that this summer will be very interesting for people who are into CSP because what we want to do is start to... when we think it's appropriate timing, start to put out whether it be whole pieces or subsystems into data for people to use, and these won't be just, you know, oh we'd like you to test it. They'll be to get user feedback on comment on implementations, you know, user interface, functionality, and to help. I mean if they're not helpful no one's going to want to even open them up, well some people will, I would have it were me because it'd be interesting. But there'll be introduction... the summer will be interesting, put it that way. That's what I'll say regarding timing.

Dave Harms: Okay.

Andrew Guidroz: Okay. Well, related to that, let's give you an easier date to hit. The problem tracking system that's part of that new program.

Bob Zaubere: Sure. Yeah, and actually just to clarify that, it is, you know, introduced at the same time as the new programs, SMP and CSP, but it can be used by anybody. You don't have to be in either program to use it. I need to look at it today. It's been done entirely using the PHP templates, which was a good thing for the templates because

obviously it causes to make some improvements. We saw things we didn't like how they were behaving.

Dave Harms: And also a nice example of how they can be used.

Bob Zaubere: True, true, absolutely. So it may be that it will go live today. It will continue to evolve, it's not going to be perfect. I looked at some of the search capabilities today, and wasn't really thrilled with them, and I suggested we need to make some improvements to it. But it may happen today, it may take some work over the weekend, it may be Monday or Tuesday, but it's imminent, and it will be I don't want to say a project, but it'll be something that will continue to evolve as people use it and they say why couldn't it, you know, do this or that. Probably within a few days it'll do this or that if it's a valid request, and we think will be useful to the community in general. So it will be something that will be... that'll be quite good, and it's imminent to be released.

Andrew Guidroz: Okay. Have you tried to put a pencil yet on the pricing of C7 or the .NET product?

Bob Zaubere: Well, in a way I have because people in CSP have guaranteed that they'll receive C7. So some of us have already paid up and own C7, and will get it...

Dave Harms: And the pricing of that suggests that C7 will not be significantly out of line with previous releases of Clarion 6. So is that...

Bob Zaubere: True, yeah, absolutely true. And as far as .NET goes, I think what we're going to end up doing with that is, the pricing isn't set, I'll tell you that right now. But what we will do is we're going to at the time that it's ready, we don't really want people to speculate on the price too much, but when it's ready we'll definitely introduce two things. One, a subscription program for .NET, and one that says I want WIN32 and .NET dual platform subscription, and some way to upgrade from the current CSP to this...

Andrew Guidroz: The universal...

Bob Zaubere:... Yeah.

Dave Harms: So I know when I saw the AVIs from the conference, my first impression was that hey maybe we'd get a handcoder's edition of Clarion.NET first and then a C7 beta, and then somewhere down the road a Clarion.NET full release. But it sounds like you're anticipating a C7 beta before any kind of a Clarion.NET release. Is that a fair

guess?

Bob Zaubere: I think probably true, Dave. I mean the... I mean certainly we could do a handcoder's version relatively soon for Clarion.NET, but I don't, you know, and I know that certain people that would be just what they want, and it would be great. And most likely probably we'll do that. But in terms of Clarion 7, it's further along down the road because it had... didn't need a compiler for instance, and it didn't need a new run-time library written in .NET. So it is much further down the road.

Dave Harms: Meaning it's more done, not we have to wait longer.

Bob Zaubere: Right, I'm sorry. Glad you clarified that. I mean further down the road than I'm on, that's for sure. <laughter> So yes, it's much closer to done, and...

Andrew Guidroz: So, I guess one other question we had with this whole thing is .NET and C7 are, although they have a common IDE, they are two totally different products. Do I have to buy C7, let's say I'm a guy coming in from the outside, you know, and saying well I'm not really a Clarion guy, I'm more of a .NET type of guy, but this Clarion thing looks interesting. Does that person have to participate in C7 in order to use your new .NET product?

Bob Zaubere: No, they don't, for sure. There's definitely, you know, without question, there will be a .NET product which is just .NET. The idea of having an IDE that can work across .NET and WIN32, and in fact cross other languages besides Clarion, is our, you know, is our intended path, that's where we're headed. So that implies right away that we won't say well you have to have Clarion 7 just if you want to use Clarion.NET. Then maybe people, for instance who are still using Clarion 2.1 believe it or not, I guess you do believe it, you know it's true. And maybe one of those would come along and say, you know, I'm ready now to move from DOS, but I'm going to skip Windows. Okay, no big deal, you can go right to .NET, skip WIN32 I should say. So yes, yeah, it'll be available as a product unto itself, and for those who need to work in both worlds, you'll have one IDE to learn, and the learning curve won't be much, but you'll have one IDE under which you can operate all of your languages, all of your platforms.

Dave Harms: And presumably it would be cheaper if say you want both C7 and Clarion.NET, the second license it'll be less expensive than if you were to buy both separately?

Bob Zaubere: True, without question.

Andrew Guidroz: People saw an assignment operator...

Dave Harms: Oh yeah.

Andrew Guidroz: ... in the AVI having colon and colon equal...

Dave Harms: Colon equal. What's up with that?

Andrew Guidroz: What's that about? Is that really coming or is that just you with testing, or what?

Bob Zaunere: Well, that exists in Clarion.NET. Now people might say what the heck do we need a new assignment operator for? And what do I have to learn now? And the answer is you don't have to learn anything because you don't have to use it. That, if you look back at the AVI, what you would find is that that is only used in the context of code coming out of forms designer. And what's the significance of that? That significance is the forms designer uses the Code DOM in order to generate code to represent a window. And Code DOM only supports, only has the ability to work with one assignment operator. Now we know in Clarion that that isn't the case, that isn't how we work. We have a simple assignment operator that assigns values and one that assigns references. But that isn't the case for the Code DOM, and the Code DOM is exactly how we generate code from the forms designer. So the := implies a reference assignment for a reference type, and a typical value assignments that you use to assign into a long et cetera. So it's different from the existing assignment operators that we have, and we needed to have just one because the Code DOM, as I said, won't work with them. So the bottom line is this, no one has to use it. The forms designer will use it, but we only included it to support the Code DOM generation, and that is how the windows structures... that's how they're created. I said structures, let me scratch that line out. That's how the representation of the window is created via the Code DOM. And that's typical, and that's used across probably, I guess I could say 99% certain every implementation of a forms designer takes that approach, it's the right way to do it.

Dave Harms: Okay. Well there was actually one other operator, somebody I noticed there, and that's... or it's actually an overloaded operator, which is plus equals when you're adding another event handler?

Bob Zaunere: Right. Well the plus equals isn't really new, and the idea of using it in that fashion is < isn't new. But if you take a look, and I'm sure you probably have, I know you've delved into .NET probably quite a bit, you'll see that that is a common way to

work with event handlers. Much of what you'll see as far as that type of change, you can look at the syntax of C#. C# being the first, I guess it was the first, well, at least the base implementation for that Microsoft uses to work with .NET. It's the most well-documented implementation of a language, of a .NET compliant language. So some of the things you'll see in Clarion.NET that are new, that were needed in order to make true use of the entire framework class library, which they said from day one that was our goal...

Dave Harms: And we're still talking about Clarion.NET being a full producer in consumer...

Bob Zaunere: Absolutely true.

Dave Harms: Yeah, okay.

Bob Zaunere: So, you'll see a few things like that that had to be added so that we could meet that goal, and they're useful things and they're not, you know, they're not major leaps. Like I said, the window designer use of the colon equals is, you know, someone could use it if they wanted to, but they never have to. They never have to use it. The forms designer did need to do it, but users don't need to do it. But the compiler will recognize it if you do use it, so you certainly can.

Andrew Guidroz: Is there anything else we're supposed to ask? I mean is that it?

Dave Harms: No, that's lots. And thank you very much Bob, we really appreciate you taking the time to talk to us.

Bob Zaunere: Thank you for giving me the chance to because certainly we're going to make some changes, and this is great to get a chance to talk with you guys, and I know that there's legitimately a lot of questions in the community, and a lot of people want a lot more information that we currently have had time to get up on the website.

Reader Comments

[Add a comment](#)

Clarion Magazine

Planet Clarion Transcript: Bob Zaunere on the new IDE and Clarion.NET

Published 2005-07-22

<Intro music>

Andrew Guidroz: This is Planet Clarion for June 30th 2005. I'm not Dave Harms.

Dave Harms: And I'm not Andrew Guidroz.

Andrew Guidroz: Good day.

Dave Harms: Good day.

<Music>

Andrew Guidroz: Well, we've had a lot of things happens since the last time we've had a podcast.

Dave Harms: We have. We've spent a lot of time waiting for some further news from SoftVelocity about a few things. And now we've had some news.

Andrew Guidroz: Right. We've got 6.2 was released along with even a maintenance hot fix type thingy.

Dave Harms: A maintenance hot fix and aside from that thingy, we've also go the bug tracker thingy. Although we're not actually going to talk about the bug tracker today.

Andrew Guidroz: Not in any great detail.

Dave Harms: No. Well not at all probably. But we just talked about it. So I guess you're right.

Andrew Guidroz: It's still new. It's all out there. Everyone just starting to utilize it so we'll go into more detail as we see the thing get used.

Dave Harms: And one of the things that happened since the last podcast, there was some stuff released from the conference in Brazil. There were some presentations that Bob and Diego had done and they were put up on the net and they caused a certain amount of confusion because there were some things there that people just didn't understand. And so we thought it would be a good thing to talk to Bob Z about this and find out actually what really is going on and ...

Andrew Guidroz: Why did this happen? How did it come out this way and what's it all about? What does it all mean for the future of Clarion?

Dave Harms: Yeah. So let's get to that right now. So we have Bob Zaunere on the line. Hello Bob.

Bob Zaunere: Good morning, Dave.

Dave Harms: And...you'll notice that Bob is on a landline. We were unable to get a proper Skype conference call going for some reason. It's still a Skype conference call but just with one landline. So hence the difference in sound quality.

Andrew Guidroz: So where ever Bob is, maybe he doesn't even have the Internet, you know what I mean? He could be like in the middle of nowhere.

Dave Harms: That could be. He's on retreat.

Andrew Guidroz: On retreat.

Bob Zaunere: They have the Internet everywhere now a days, Andrew.

Andrew Guidroz: I like the fact that you still refer to it in the singular rather than like our US President who refers to it in the plural. The Internets.

Bob Zaunere: Well maybe he's thinking about Internet 2.

Dave Harms: That could be it. Let's give him the benefit of the doubt. All right, well we wanted to...

Andrew Guidroz: Ask a million questions.

Dave Harms: Ask a million questions. First of all, thank you very much for talking to us and there has been an awful lot of discussion about Clarion 7 and Clarion.NET ever since the presentations went live. The presentations that you did for the conference in Brazil.

Bob Zaubere: Yes. Lots of fun.

Dave Harms: Yeah.

Bob Zaubere: I'll make a comment about that just to give perspective. Those presentations, they caused a lot of fun. I enjoyed reading the wild speculation quite a bit when I had time. The fact is that the presentations weren't supposed to be released because they had no context under which they were released. So prior to the presentation that was released was a lengthy explanation of what we were showing and why we were showing it and the intent of it. So it was interesting to see when people didn't have the context how they came up with their own interpretation of what they were looking at and what it meant.

Andrew Guidroz: The proper phrase for that in the community is we call it like we see it and if we don't see it, we make it up.

Dave Harms: Do you want to just lay out the context briefly for those presentations?

Bob Zaubere: The context was really this. There is some work being done on an IDE in order to incorporate some new functionality that we needed for the .NET side of the product line. And we chose to show that because it was brand new and interesting. What we didn't show and what had no context because there was no introduction for me to explain it, was the fact that that was just part of the implementation for .NET. It led to people believing that .NET could come out before Clarion 7. Clarion 7 is nowhere near done. They're just starting with an IDE. Other kinds of crazy things. But the fact of the matter is that was simply-- we simply chose to show that because it was brand new and it was interesting. And we thought this will be fun to show how much we've got done on this side of the product.

Dave Harms: So it wasn't really intended as a snapshot of where SoftVelocity's at these days in general.

Bob Zaunere: Not by a long shot. And the only thing that you could say in that regard is it was a bit of a snap shot of how far we are along with the .NET part of the project. But, you know...

Andrew Guidroz: So that won't be the same IDE used for the C7 project?

Bob Zaunere: That IDE, part of that IDE will be used. There will be a common IDE. There's no question about it. But certainly, there's two things that are true, Andrew. We had two IDEs internally. We have one that is 100% Win32 and one that is not 100% Win32. Which is the one we showed. Which is a mixture of both. Really, what we've done is we've grafted together the functionality in the two IDEs to get one IDE that will support everything that we need for both products.

Dave Harms: The IDE that you showed is pretty clearly based on SharpDevelop and we talked about that a little bit last time as well. Do you want to say a little bit about how you came to decide on getting a license for SharpDevelop to use that as a basis for the front end?

Bob Zaunere: SharpDevelop, the SharpDevelop IDE, we looked at it a long time ago. It has a lot of the functionality that we intended to implement. That we needed to implement in order to properly support Clarion.NET as a language. For example, the capability, the simplest example, a forms designer that works with .NET controls. Now we had one of our own that was pretty far along. But it wasn't as complete as what was in the SharpDevelop IDE. In any event, until you've actually written an IDE, it may appear that an IDE is a frame with a bunch of things plugged into it. And that simply isn't the case. An IDE really is a pretty complex beast underneath the hood. Because there is a lot of communication that has to be handled between the applets or components, however you want to characterize them, and the plumbing, so to speak, is the critical part of an IDE. The UI is not. It's the plumbing underneath. Now we obviously have a long history of having a pretty extensive IDE. And that IDE has its own plumbing. What we found when we actually looked at the source code for SharpDevelop is that the architecture was quite similar to what we had. If that hadn't been the case, we probably wouldn't have gone forward at all with SharpDevelop because it would have lost time instead of gained time. But because it was so similar and philosophically was along the lines of how we think an IDE should behave, we found it a great match.

Dave Harms: And that's actually a really interesting point because I assumed that you were able to look at it because you can download the source because there is an open source edition, and you could examine that without having to, say, enter into some sort of

negotiations, doing an NDA so you could look at the source code.

Bob Zaubere: True.

Andrew Guidroz: So you seem to be implying, and I'm just going to clarify this thing, Dave's thinking one way and I'm thinking a different way and I'm sure there are a million different opinions on the news groups. You've decided that although this thing has its origins in an open source product that some people utilize from that way and just plug into it, you've actually taken a different approach where you've gotten a copy, a license to get the actual source code, and you're going to actually fork the product on day one and modify the source code.

Bob Zaubere: [pause] Was that fark the product?

Andrew Guidroz: That's correct. Yes.

Bob Zaubere: You're going to have to...

Andrew Guidroz: F, O, R, K.

Bob Zaubere: Oh. Okay.

Dave Harms: With the Cajun pronunciation.

Bob Zaubere: Got it. [laughter] I thought it was F, A, R, Q. I was just like now you threw one at me that I don't know. Okay. I'm with you.

Andrew Guidroz: Fork.

Bob Zaubere: The source code we knew from the beginning was going to change dramatically because we certainly never would have, well I guess not obvious, but we never had any intention to throw out our own IDE and all of the intricacies that we have in place and just re-implement them. So the SharpDevelop IDE, those guys are very talented people. It's an excellent architecture, in my opinion. But it's designed for what they needed to accomplish with their IDE. And their IDE is obviously one designed for hand coders. Our IDE obviously comes from a different point of view entirely. A lot of people do write hand code and Clarion's a great language for that, but really our IDE is about application generator and the dictionary and the templates and what goes with it. So,...

Andrew Guidroz: You see where I'm going. What I'm curious about is, let's say they're on 1.1. today and tomorrow, they release 1.2. There's no such thing as me going to the SharpDevelop site, downloading the latest version of the IDE and continuing to roll from there. I'm literally going to have to go through you for the distribution because you've actually modified the source code and you're the source code keeper from that point forward.

Bob Zaukere: That is accurate, true.

Dave Harms: And actually I think, maybe just to clarify, as I understand it SharpDevelop, the version that you can go and download from your site for free and it is released under the GPL, the GNU Public License, and the GPL is pretty restrictive, I think. And if you were to create a plug-in for SharpDevelop, I suspect that your plug-in would fall under the GPL. Isn't that right?

Bob Zaukere: Yeah. I believe that is true. I think within the context of SharpDevelop that they probably have the option to allow plug-ins that maybe are not GPL. But I'm not sure about that.

Dave Harms: Yeah. Well I think there's also the LGPL which I think makes that kind of thing a little bit easier. But the GPL is actually a pretty restrictive license. So the other point there being that because the SharpDevelop folks have developed everything from the ground up, they have the right to distribute one version as GPL and another version as commercial source codes, say, to SoftVelocity.

Andrew Guidroz: Just like MySQL does.

Bob Zaukere: Yes. Absolutely true.

Andrew Guidroz: Because that's exactly what Amazon did, right? Amazon went out originally, I'm not sure where they are today, but they originally used MySQL by buying a copy of the software and they modified it from there. That was the original deal. I've used MySQL for some products just because of what I saw Amazon doing with it.

Bob Zaukere: That's interesting. I didn't know that.

Andrew Guidroz: Well then, people saw an IDE when they went to DevCon.

Bob Zaukere: Right. And really the explanation that wasn't heard, was that we're going

to show you some of the functionality that will exist in the new IDE. Not what the final UI will look like, certainly. Obviously we hadn't plugged in AppGen or Dictionary Editor, but that's an aside. The UI itself will change. But some of the functionality was there. And we wanted to show the progress on the .NET compiler and talk about some of the changes in the language etcetera. So it was an hour long presentation. We couldn't cover everything. I had intended to show that we've got the support that we talked about in the last podcast to do Win32 compilations with various versions of Clarion so that I could bring out the Clarion 7 Compiler, Clarion 6, Clarion 5 etcetera.

Dave Harms: The current version of Clarion 7 that you have, now this IDE, you can do Win32 compiles, right? You can do both Clarion.NET and Win32 and VB.NET and C#.

Bob Zaubere: Yes. Absolutely true. That was what I did not get to show during that hastily created AVI but what I had intended to show, had I gotten my visa in time, was to start off by saying "Okay, lets go through," and I had a copy of 5.5, 6 and 7 by the way on my system and I was going to say "Look. Lets bring up this and compile it with 7. Lets bring it up with 6. Lets compile it," and a couple other little tricks that I was going to do. But didn't get the chance to do it. It was a big time constraint.

Andrew Guidroz: This running multiple versions, you find it was made easier by going to this new IDE because it's not Clarion. It doesn't share common runs, DLL and that type of thing?

Bob Zaubere: Well no. Actually Andrew, the architecture to do that was plugged into the next context but it existed in the old IDE. It isn't so much the dependencies because what happens, what we calculated a long time ago, is that the project system itself has to be aware of the version. So you get the registered versions with the IDE. And then it has to find the right versions to link in. It has to know things like for instance, if you say "Okay, I'm working with Clarion 6," and you say "Add the TopSpeed driver to this project," it has to know "Okay, he's on Clarion 6, lets add C60TPXS.dol," as opposed to C70 etcetera. So that part didn't change. We already had that code done.

Andrew Guidroz: So literally, your own IDE that you were doing from scratch, you're not going to use it but you're going to use functionality that you were writing into it.

Bob Zaubere: Well that's sort of correct. The IDE, the internal IDE is separate from SharpDevelop. It will continue to be developed. What we didn't do in the older IDE...

Dave Harms: Wait wait wait. Hold it there. Why would you continue to develop an old IDE if you have the SharpDevelop IDE that your using as the basis now?

Bob Zaubere: Well, “develop an IDE,” maybe it’s too vague a return. We’re talking about two different things here though, aren’t we. We’re talking about Win32 and we’re talking about .NET. The kind of subsystems that I told you, we won’t do away with. And I’m not saying like the bigger ones like AppGen. There are ones that people aren’t even aware of like the whole ASL layer.

Dave Harms: ASL meaning application software...? What does ASL mean?

Bob Zaubere: Support layer.

Dave Harms: Application Support Layer.

Bob Zaubere: Applet Support Layer.

Dave Harms: Oh. Applet Support Layer. Okay.

Bob Zaubere: Sure. Look at any distribution of Clarion going back to C4, maybe CW2, but you’ll find a C??ASL.DLL. And you’ll find other files and you’ll go “Well, I don’t use any of those.” Well, of course you don’t. They’re part of the IDE. When you look at the new IDE, if you look at it deep enough, the one that we showed in that little AVI, you’ll find that the C70ASL and all it’s associated components are part of the IDE I showed. And so what we do, is we will continue to plug things into that and expand on that. Because that is [what], when they’re ready, will be plugged into what will make the final IDE.

Dave Harms: Yeah. That’s interesting. So if you can summarize the overall architectures here, we can think of SharpDevelop as sort of the layer on top and underneath it, you really sort of have two major sub-systems. You have the Clarion.NET sub-system. Oh no. this is confusing me now.

Bob Zaubere: It’s a complex topic. That’s why.

Dave Harms: Yeah. But I guess you’ve got the major pieces you’ve got there. You’ve got a huge sort of what we’ve just been calling the old IDE which actually will still have sort of the SharpDevelop skin on it. Right?

Bob Zaubere: Well actually, it will for a little bit. But the UI is going to change. There’s not much question about that.

Dave Harms: Okay so you bought the SharpDevelop license really more to assist with the Clarion.NET end of things than to assist with the visual appearance of things per se?

Bob Zaunere: Yes. I mean, it is true. Again I go back to this point. An IDE, some people may think it looks pretty simple. I see a lot of people saying "I could port the existing IDE in two weeks." [laughter] It's nothing. Flip a switch and recompile it. It's crazy really. Even if you look at what goes underneath any IDE, you'll find out an IDE is a very complex piece of code. Think about Visual Studio 2005. It's been in beta for a couple of years. Maybe longer. Being tested by everyone and they're still working on it. And it isn't that dramatic a change from Visual Studio 2003. What could be going on? Well I'll tell you what. It's an extremely complex piece of code, an IDE is. So people kind of had the wrong perspectives in that regard. The pieces of SharpDevelop that we, you know, they have an excellent support layer for working with .NET. That they do. And they can work with Win32 code to. I don't mean to imply that they don't. But we have an extremely well developed system for working with Clarion and Win32 and AppGen and Dictionaries and the like that we have present now. All of the code needs to exist to go forward, or it has to be re-written. Which of course, we are not going to do. Licensing the SharpDevelop IDE will save us hundreds upon hundreds of hours. There's no question. But it isn't what people appear to think which is boy, all they have left to do is just plug a couple of things in and I could do that this weekend.

Dave Harms: So does that mean that then there are kind of two major areas that are developer related to the IDE. It seems to me, part of it is certainly integrating the existing IDE code into the SharpDevelop framework. And then there's also going to be sort of another major step of marrying the current IDE AppGen, Dictionary, Editor stuff with the whole .NET side of things. Is that a fair statement?

Bob Zaunere: Yeah. Sort of. You know, it's not so much integrating because it almost sounds like plugging in and I use the term grafting because maybe it implies something a little bit different than integrating. But in an IDE there is a lot of inter-communication between various components. And it may appear on the surface that, take our existing Clarion 6 IDE. Maybe it looks like on the surface that there's just a handful of components. There's Dictionary and there's AppGen and there this and that and the other thing. But really that just isn't the case. Underneath each of those are major or rather, significant subsystems. And some of them are designed just for communication between other Applets and some of them have other purposes. The same is true in the SharpDevelop IDE.

Dave Harms: And I guess also the way you're describing the whole architecture of this thing, it also suggests, you know, people have said "Well, why don't you create a plug-in

for Visual Studio,” or something like that. And from what you’re describing to me, it sounds like...

Andrew Guidroz: It’s more complicated than a plug-in.

Dave Harms: ...it’s more complicated than that and a plug-in would not enable you to have sufficient functionality. Is that right?

Bob Zaubere: Mostly it’s right. There may be a day when we’ll say “Okay, let’s create a plug-in for Visual Studio.” But it won’t be today because I don’t think that’s what our users want. Visual Studio isn’t inexpensive in and of itself. Writing a plug-in for Visual Studio can definitely be done. We may do it, but by the time we would do it, it’s a lot different from what other people write for a plug-in. Other people who write plug-ins for Visual Studio are basically saying “Okay, I’m going to support maybe a compiler” if they have one, support working with the Visual Studio Editor and Projects system. There’s a few other bits and pieces, but essentially that’s it.

Dave Harms: Which is a lot simpler than something like the AppGen environment.

Bob Zaubere: AppGen, Dictionary, Round Trip Code Generation etcetera. It’s a different beast completely. Because the Visual Studio IDE is for handcoders. And we look at it. They like to look at our product, we know that for a fact. And we like to look at theirs too now. So there’s a few things in Visual Studio 2005 that are pretty nice...

Dave Harms: Sorry Bob. Just to go back, you said they like to look at your product. What does that mean?

Bob Zaubere: Well there’s actually people who work over at Microsoft on the Visual Studio team who are former Clarion users. I’m not going to name any names. Who were very much influenced by what they liked about Clarion.

Andrew Guidroz: Some of the guys directly involved in the C# part of the product you mean, right?

Bob Zaubere: With the Visual Studio team specifically.

Andrew Guidroz: Okay. All right.

Bob Zaubere: But anyways. That’s an aside. So if anyone...

Dave Harms: That's a very interesting aside. It's interesting. There is that story I remember Bruce Barrington telling a long time ago about Bill Gates assembling some of his vice-presidents and dropping, what, a half dozen copies of Clarion Professional Developer on the desk and saying "Tell me why these guys can do what we can't do."

Bob Zaunere: Yep. I remember that story as well. I'll just go back to the point that because that AVI was released when it shouldn't have been released because it has no context, it wasn't a marketing piece. It was just a short technical demonstration that before it had a lengthy introduction of what we were going to do and what we had been doing. So it took it out of context. But I understand people's point of view as far as the IDE goes. And some people felt great about it. I didn't read every message. There were way too many. But mostly, I saw very positive feedback and a few who completely misunderstood and literally thought "There's no AppGen any more. What is that guy doing to this product?"

Dave Harms: Can we ask you just a couple of more specific questions about individual features. Regarding the debugger, in SharpDevelop I guess the 1.1 version doesn't have a debugger. The 2.0 has an integrated debugger. Presumably the debugger for Clarion will still be the Soft Velocity debugger. What about the Clarion.NET stuff. Will that be your debugger? Will that be a Sharp 2.0 debugger? What will it be?

Bob Zaunere: Well there's a pretty well defined way to debug .NET programs because under the covers there's all IL code. So internally we have, and had for a while, the beginnings of a .NET debugger. It certainly will be different from the Win32 debugger. It needs to be. It's entirely what is being debugged underneath the program. Now whether we do anything with the SharpDevelop implementation, it's too early to say right now.

Andrew Guidroz: How bout COM controls. Are we going to be able to see these things active and working in design time whenever we're doing things in the IDE?

Dave Harms: Yeah. That seemed to be suggested in the AVIs but I wasn't quite clear on that either.

Bob Zaunere: There's two contexts to look at that, and one decision hasn't been made. So I don't know that I can give you a firm answer. There's two ways for us to go forward. We have the existing screen formatter. It works as Win32 with Clarion implementations of Win32 controls. And now we have the choice of using that with Win32 or doing further work on the forms designer so that we can properly represent Win32 controls in it and use it for Win32 code. It's quite different from what people might imagine, I would bet.

Dave Harms: The forms designer is different or the process is different?

Bob Zaunere: The processor is entirely different. The process by which the controls in general are worked with under .NET is one hundred thousand percent different from how you work with them in Win32. By that, as an example, the form designer uses reflection to discover everything about a control and it only discovers what the control exposes. If you were to look at almost any component, any Window control, like a button for example, and you look at what's available in the .NET implementation versus what you might find in the current Clarion button, you'd find a lot of overlap and you'd find a lot of discrepancies. Some things that over the years our users asked us to add are present when you look at the control properties for our button, and some are not. And so therein lies the undecided fact that we have two paths to choose from. And one would be to use the forms designer and implement an entirely different way in working controls in it. And the other would be to use a Win32 forms designer, as we have now, within the new IDE.

Dave Harms: And that raises other questions as well. Maybe questions of migration. I mean migration from Clarion 7 or from Win32 to .NET, you're taking an AppGen [app] over because if the forms designer has different capabilities and the controls have different capabilities, what kinds of circumstances might arise when you're trying to migrate an App across?

Bob Zaunere: Well, it depends really. We're very fortunate in one respect. I say "We" meaning Clarion developers in general because, as you know in an App, controls are stored within AppGen. So they have a descriptive representation of the control and its properties. And what that means for us is that that descriptive representation in AppGen currently, is transferred into a Windows structure, for example. But it could just as well be transferred into, or rather, de-serialized into representations that we showed in the AVI. Which is a standard way that Win Forms uses to describe a control.

Dave Harms: Right. And that's an advantage that Clarion has over maybe other languages because there basically is all the metadata.

Bob Zaunere: This will be a controversial issue and I did mention this in my pre-cursor to the AVI. But right now in Clarion.NET, we have not implemented the WINDOW structure. And there's two reasons behind this. I mean, with the WINDOW structure obviously goes the ACCEPT loop. Those two go hand in hand. But I'll tell you the reasoning...

Dave Harms: And it appears to me that there is no ACCEPT loop in Clarion.NET at this

point. Is that right?

Bob Zaunere: At this point, there isn't. Our initial goal as far as that goes is that we wanted to initially make sure we support working with .NET the way the world works with .NET. As opposed to say "You know, we have a better way and we're going to do the .NET way later." We took the opposite path by deciding to do that. That we should say "Okay, this is how Windows are worked with in .NET and this is how we'll do it in Clarion." Now that's what we saw during the AVI. Again, when you come from an App file, it won't even be noticeable. There's no one out there, I believe for most parts...

Dave Harms: Sorry. An embed point with except loop is really not significantly different from an embed point in an event handler because at this point, we're mostly using TakeEvent handlers anyway. Right?

Bob Zaunere: Exactly true. And by the same token, if you look at the representation for a window under .NET, okay it's not a Windows structure, but it's still pretty clear to read. However, how many of us frequently sit down and take a complex Windows structure and start editing it as text? I know I don't. Unless it's like one little attribute I want to change and I just happen to have the cursor near there and I say "Okay. Let me just type it in there." Typically we go through the forms designer. We don't work with the WINDOW structure directly. Especially with a complex window because it just gets too big. Too unmanageable. It takes longer than just popping open the window within the form designer. So, in any event, the WINDOW structure right now isn't implemented. One of the reasons behind it is what I first said. Because we intend to, with Clarion.NET, follow the path that all .NET compliant languages should support which is 100% integration with the framework class library. But the other reason is this, and this is maybe of more interest. One of the things that is true currently about the WINDOW structure and its relationship with the Clarion RTL and the compiler and the whole Clarion subsystem is that it is very convenient. It's nice, we have a structure, but it has certain limitations that imposes upon us. And they go as follows. To add any new control or to modify an existing control requires a lot of changes. It requires changes to the parser for the WINDOW structure. It requires changes to the form designer itself. It requires changes to the compiler and requires changes to the RTL. Because what happens at run time is that the WINDOW structure at compilation time, the WINDOW structure process, at run time, that binary representation is past the RTL and then it then has to use it to create the window. So in any event, my point being is that there's a lot of complexity and a lot of code that has to change every time you want to work with a new control.

Dave Harms: I was going to say, as opposed to .NET where because everything's based on a class library, it's a question of just instantiating your class and being able to use

reflection to determine what the class' capabilities are and so forth.

Bob Zaunere: Yes. Exactly. Basically, the forms designer can load up any control. Whether it's a user defined control or a third party market control or some new control that gets added to the framework class library, it will be able to load it and expose whatever properties that control says "Here. This is what I have for you, take it away." And it will build a property page based on it. So it's quite different and a lot more flexible.

Dave Harms: And given the difference is there, what would be the upside to porting the current designer and using that instead of the .NET forms?

Bob Zaunere: Well no. I never said that. If we were to port the current designer, it would be for use with Win32.

Dave Harms: Okay. It's a question then of whether you take the Windows forms designer and adapt it back to Clarion 7 or whether you use the Clarion 7 version of the current designer.

Bob Zaunere: That's correct. So it isn't quite the same beast. And therein lies why I've chosen not to implement that right now under .NET. Now, we did investigate it.

Dave Harms: Sorry. You mean not to implement the Clarion...

Bob Zaunere: The WINDOW structure.

Dave Harms: The WINDOW structure. Yes.

Bob Zaunere: Particularly, yes. And we did investigate it and it could be done. And we even came up with a way that would allow a WINDOW structure to be, how should I put it. Not quite as dynamic as what you can do right now under the forms designer, but a way for it to be extended. However, I'm not sure whether we're going to pursue that.

Dave Harms: Well I guess the benefit would be some sort of backward compatibility. It might make it easier to port stuff across to Clarion.NET. But it sounds like then probably in fairly short order, you would get people complaining about why isn't this feature there and why isn't that feature there.

Bob Zaunere: And why can't I use this third party component and how do I do it. I don't think the migration is an issue. Even if we were to talk about people whose applications

don't exist within an APP file, if they're completely hand coded, there still is a clear cut path to move from a Windows structure to a representation under WinForms. Now we haven't written any code to do it, but we looked at it. And sure, we could do it. We could write a parcel that would parse the structure.

Dave Harms: It's a fairly straightforward mapping.

Bob Zaunere: Yeah. Absolutely. So I don't think there's much of a migration issue there. I think it may be more of a psychological issue. People may think it implies something horrible. "Where's the WINDOW structure?" But it isn't actually a migration or a functionality issue except there's a plus on the functionality side. But it's not really a migration problem like "Oh, my App won't work because of what happened to my WINDOW structure." That simply isn't going to be the case. Like I said, for App files, it's basically a non-issue. And for hand coders, there's a clear cut path to get there.

<Tape rewinding noise>

Dave Harms: Okay. Well let's pause it right there. Bob did have some more to say, but we're kind of running out of time here.

Andrew Guidroz: You always do that to us. Just when I'm really getting into it. You know the next thing was going to be about the super-duper 128 bit version of Clarion.

Dave Harms: Yeah. The 128 bit version that runs on Solaris.

Andrew Guidroz: My new Solaris laptop.

Dave Harms: Your new Solaris laptop. Right. With the dual boot Mac/Solaris machine.

Andrew Guidroz: Right.

Dave Harms: Yeah. With the Intel processor. Well there was a lot of interesting stuff there. And I would imagine what's going to happen now, is that there will be, as Bob at one point said, there's some controversial comments and I'm sure there'll be a little more. I'd say, probably for every question that's been answered, there's probably another question that's been brought up.

Andrew Guidroz: Well the real thing that jumped out at me in the whole thing is the idea that Clarion wouldn't exist well within a pure plug-in play. That's an amazing thing that

jumped out at me. The fact that for the IDE to work well, it has to be integrated tightly with the IDE. That it's just not something you're going to plug in to, like, Visual Studio.

Dave Harms: His comment about that other people who are doing that are basically doing hand coders stuff. So what the [Visual Studio] IDE really provides you is it provides a hand coder's environment. It doesn't really provide you some place that's amenable to an AppGen kind of environment.

Andrew Guidroz: Exactly.

Dave Harms: He did say it could be done. Obviously, it would be an awful lot of work to do that. And then if you're doing all of that work, my guess here, not something that Bob said, but just that I'm saying, is that you're not really at that point, leveraging the IDE so much, say the Visual Studio IDE as much as you might like to.

Andrew Guidroz: You may not get the productivity. Now don't get me wrong. I think if you're used to using some other IDE and this is one more thing to plug in to, it certainly gives you much more market out there.

Dave Harms: But you have the additional cost as well, as he pointed out. So that raises the bar. Are people really wanting to pay to get Visual Studio and a plug-in? The plug-in you generally tend to think of as being worth less than the IDE.

Andrew Guidroz: From my stand point, the way he's going, it's better for me. I'm more interested in having my Clarion environment independent of Visual. If in the future they decide to say "Let's go ahead and make a plug-in," or take the portions of the existing product now and use it as a plug-in, that's cool. I may have a use for it as well. But my primary use is using Clarion within its own IDE and its own integrated very smart environment.

Dave Harms: Yeah. And really it made sense the way they did it because they – by picking up a license to the SharpDevelop code, they really got a head start on the .NET stuff. And it also sounds like it's a reasonably good fit with the architecture of the existing IDE. So they can plug all those pieces in and make it all work together.

Andrew Guidroz: Yeah.

Dave Harms: One of the other things I was starting to say before is that maybe, although I'm sure this podcast will raise a few more questions, maybe people should kind of take a

lesson from the whole AVI episode and say okay just because maybe something was said that makes you think that 'A' or 'B' is true, or maybe you haven't heard about 'C' yet. So I would say just not for anybody to get too bent out of shape if they hear something that they think is alarming, because we know that Bob Z's been a Clarion guy for a long, long time now and I think he's fairly well attuned to the kinds of things that people really value about the product. So I don't think we have to worry that something that everybody thinks is really important is suddenly going to go missing.

Andrew Guidroz: Exactly.

Dave Harms: Okay. Well as I say, we'll have more from Z on this stuff at a later time. But for now I think ...

Andrew Guidroz: That's about it.

Dave Harms: Well no, actually, there's one more thing I wanted to talk about.

Andrew Guidroz: That's right. You get to sell stuff again.

Dave Harms: I get to sell stuff. There are three new books that I want to talk about. One is the Clarion 6 Tips and Techniques book which an all-new book. A new collection of articles of Clarion 6 related stuff from the magazine. And just jam packed with things. There's a lot of really wonderful articles in there. So naturally, I highly recommend that book to everybody. And then beside the Clarion 6 book, I also have two books which are really a re-issue of the original Clarion Tips and Techniques book. And one thing that happened after I came out with it, after I started working the C6 Tips book, I realized that the first tips book didn't really have anything in the title indicating which version it applied too. And also, I've gone for a slightly new format. About 400 page book instead of the 670 page monster that the tips book was. And so I've split the tips book – the original tips book, up into two volumes. Clarion 5.x Tips and Techniques Volume 1 and Volume 2. And so that is the same material as was in the tips books. So just be aware of that if you already have the tips books and were thinking about those. However, the Clarion 6 Tips and Techniques book is in fact all new material. So, just keep that in mind. And also, these books, all three of these books will be available maybe by mid July, maybe a little later, through regular bookstores as well. So you can buy it at Amazon and you can also call your local bookstore and say "I'd like these books. Can you get them in for me?" And they should be able to because these are being distributed through Ingram which is one of the largest book distribution channels in the world, I believe. So, there you have it. Three new books. Three cool books. And also I'll just mention that the Clarion magazine summer break schedule, we take a little break from publishing at some point in the summer. This

year it's in July. Actually, I will be online periodically during the month of July checking email and so forth and there will be a couple of articles appearing in July, but basically June and July are sort of a combined month which is what we did last year as well. And that also means that any fax orders will not be taken care of until I get back. And there will probably be a little bit longer delay in PayPal orders and also in when I print the books. Instead of being able to do it every day, I'll probably be printing out book orders once or twice a week. So a little bit of delay in some of the usual stuff there, but otherwise everything should continue. If you buy your subscription with credit card online, your renewal or your subscription is processed immediately and you have access right away. So there will be no change in that.

Andrew Guidroz: Is that everything?

Dave Harms: That's all I've got. Did I forget something?

Andrew Guidroz: Well not knowing all about geography and politics, we can at least send out a happy Canada Day tomorrow ay, to all you Canadians.

Dave Harms: Yeah. And a happy 4th of July to all you 'Murricans.

Andrew Guidroz: Murricans.

Dave Harms: Canajans and 'Murricans.

Andrew Guidroz: There you go. Canajans.

Dave Harms: And I guess a happy Deep Impact Day.

Andrew Guidroz: The Deep Impact. Cool. That's July the 3rd, right?

Dave Harms: I though it was on the 4th.

Andrew Guidroz: I think it's going to actually end up being the 3rd the way it's going to fall.

Dave Harms: Somebody didn't calculate the numbers right. Well if you don't know, go look it up on the web.

Andrew Guidroz: Yeah. Go look it up.

Dave Harms: All right. I think we're done.

Andrew Guidroz: Take care. Have a good day.

Reader Comments

[Add a comment](#)

Clarion Magazine

Online Mapping And Routing In Clarion

by Colin Wynn

Published 2005-06-22

I develop an application for the UK Removals and Storage industry, and felt that adding a facility to map and/or route the addresses held within my database would be a nice enhancement to my application.

Now I had a few "routes" I could take, interface with Microsoft Autoroute, interface with Microsoft MapPoint or interface with the web. I chose the latter, because I didn't want to force my users to spend money on software my application requires, and I figured that websites would be kept more up to date. The downside is that an internet connection is required, but it's my belief that if you're in business today and don't have internet access, then shame on you!

I set about looking at the various websites around, including Google maps, Multimap, Streetmap, and MSN maps. I settled on the latter. I like [MSN maps \(UK\)](#), not only because it's powered by Microsoft's MapPoint Technology (so those users who have used Autoroute in the past will be used to the style, detail and quality of the maps), but more importantly because of the user interface. When mapping and routing information passed to it doesn't quite make sense, MapPoint prompts the user for further input where necessary.

Now that I had chosen the website I set about implementing it in my application. Well, in my application I used CapeSoft's FileExplorer, but in the downloadable example I went with the article [Using The Web Browser OCX](#) by Matt Grossmith published back in 2002.

The example application provided is a database of football teams, and I store the team name, the stadium name and the address. Within the application you can choose to map the address of the team stadium using the map button, as shown in the bottom left hand corner in the window in Figure 1.



The image shows a Windows-style dialog box titled "Record Will Be Changed" with a close button (X) in the top right corner. The dialog contains a form with the following fields:

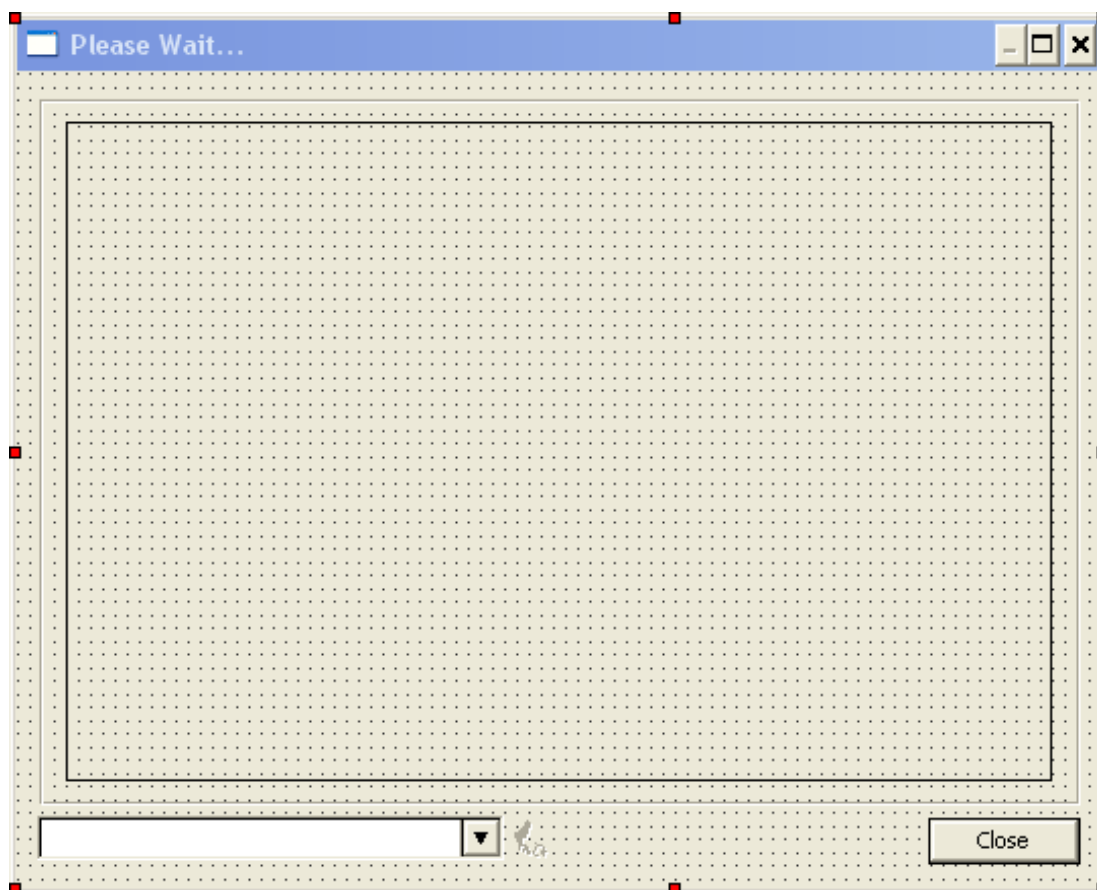
Team:	Aston Villa
Stadium:	Villa Park
Street:	Trinity Road
Town:	Birmingham
County:	West Midlands
Postal code:	B6 6HE
Country:	United Kingdom

At the bottom of the dialog, there are two buttons: "Save" and "Close". A small globe icon is visible in the bottom left corner of the dialog's content area.

Figure 1. The map button on the address form

When the map button is pressed I call a procedure (JourneyMapper) where I pass the Team, Stadium, Street, Town, Postal Code and Country, all by address (rather than by value). Passing the address data in makes this procedure completely reusable elsewhere within my application.

JourneyMapper is simply a window. Upon the window I have placed a Group Box and within I placed an OLE control. I also placed a FileDropCombo and a Button, but more on those later. Lastly a close button, thus the window formatter looks like Figure 2.

**Figure 2. Creating the JourneyMapper window**

Once the JourneyMapper procedure is called I need to populate a map of the address passed onto the window. I do this by building up the call to MSN's mapping URL in the Open the Window embed. Firstly I have to set the domain name to use (this can be country specific and is therefore stored in the Defaults file of the example), and then add the call to the correct .aspx page –there are two, one for mapping and one for routing. I also have to add to my URL details of the address I wish to map, and I can supply details such as street, city, state/province, zip/postal code, country. So I build up my URL as follows:

```
?URL{ 'Navigate(URL=" ' & |                               ! Build mapping URL
      Def:URLDomain & |                               ! URL domain name
      '/home.aspx?' & |                               ! .aspx to use
      'str1=' & p:Street & |                             ! Street of origin
      Choose(~p:Town, ' ', '&city1=' & p:Town) & |       ! Town of origin
      '&zipcl=' & p:PostalCode & |                       ! Postal Code of origin
      '&cntyl=' & Loc:MSNNumber1 & |                     ! Country Number of origin
      ' ', Flags=14) ' }
```

And that's it! With the team in Figure 1 selected I get the map shown in Figure 2. The address is marked by a pushpin

graphic.

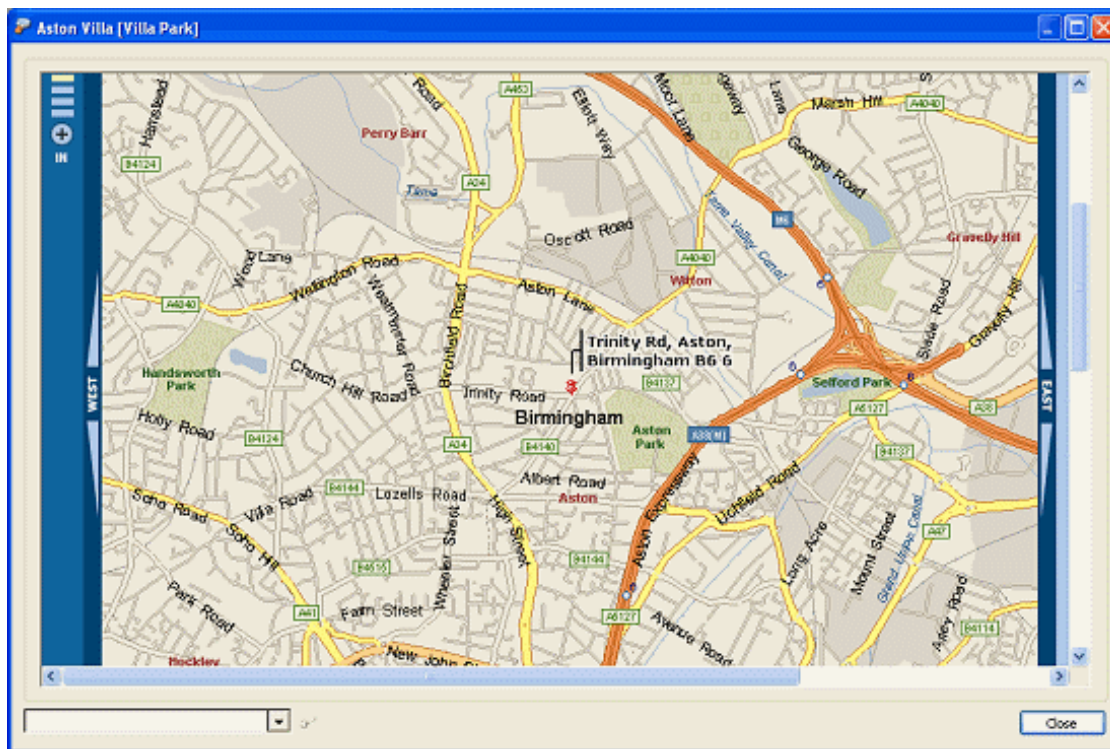


Figure 3. Map showing location in center ([see full size image](#))

To add a bit more complexity on the window I added a FileDropCombo control (a file alias of Team). Selecting another team enables a journey button, allowing a route to be generated from the origin team to the destination team (the team selected from the FileDropCombo).

Once a team is selected and the Journey button pressed I simply build a new URL, passing not only origin address details but also destination address details, and this time I call a different .aspx page. I build up my URL as follows:

```
?URL{'Navigate(URL=" ' & |                                     ! Build routing URL
  Def:URLDomain & |                                           ! URL domain name
  '/directionsFind.aspx?' & |                                   ! .aspx to use
  'str1=' & p:Street & |                                         ! Street of origin
  Choose(~p:Town, '', '&city1=' & p:Town) & |                 ! Town of origin
  '&zipc1=' & p:PostalCode & |                                   ! Postal Code of origin
  '&cnty1=' & Loc:MSNNumber1 & |                               ! Country Number of origin
  '&str2=' & ATeam:Street & |                                   ! Street of destination
  Choose(~ATeam:Town, '', '&city2=' & ATeam:Town) & |       ! Town of destination
  '&zipc2=' & ATeam:PostalCode & |                             ! Postal Code of destination
  '&cnty2=' & Loc:MSNNumber2 & |                               ! Country Number of
destination
  '&rtp=' & Def:Shortest & |                                     ! Routing type
SHORTEST/QUICKEST
  '&unit=' & Def:Kilometers & |                               ! Unit type KILOMETERS/MILES
  '"', Flags=14)'} }
```

Note above that you can even pass whether you want the shortest or quickest route, and whether you want the unit to be calculated in kilometres or miles (stored in the Defaults file within the example).

So, choosing the mighty Juventus I get the route map in Figure 4.

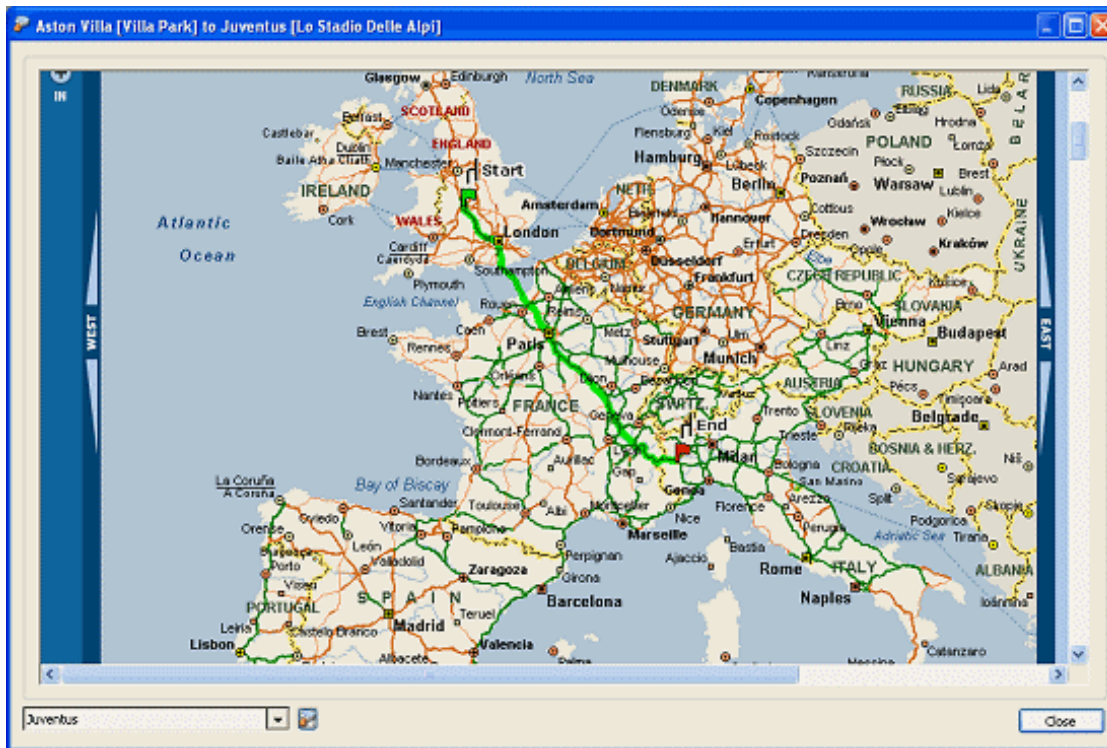


Figure 4. Map showing route ([see full size image](#))

One good feature of MSN's maps is the routing detail, as shown in Figure 5, and the way you can click the checkboxes and get a turn-by-turn map.

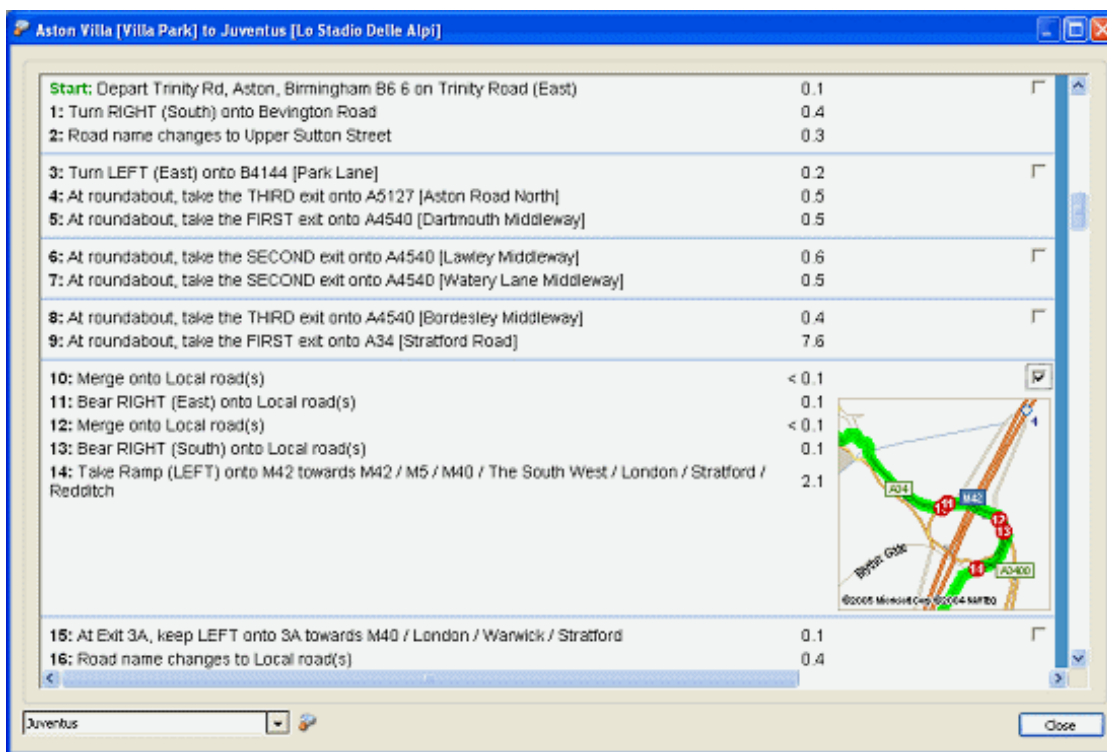


Figure 5. Routing detail ([see full size image](#))

There is a lot you can do with maps within your applications, and Microsoft's [Building Links to MSN Maps & Directions Maps](#) page is very useful indeed.

[Download the source](#)

[Colin Wynn](#) has been using Clarion since 1990 and runs his own software house in the UK. A former TeamTopspeed member, Colin used to run the largest Clarion-related bulletin board in Europe, and he was the president of the UK Clarion User Group in the early 90s.



Reader Comments

[Add a comment](#)

- [» Very nice Colin! A real simple solution, gotta like...](#)
- [» Excellent Colin, a great real-world example!](#)
- [» Very nice, I'll use it. I did something similar using...](#)

Clarion Magazine

A Multi-Threaded Queue Of Queues

by **Svetlana Zusman**

Published 2005-06-21

If you write programs using Clarion, chances are that you directly or indirectly use queues. What can be simpler? By declaring a queue you can add, access, modify, delete and sort records without worrying about how the memory for the queue data is managed, or how access to stored data is achieved. Queues let you store, organize and provide fast access to your data within the program.

You can even have a queue that contains other queues. But what if multiple threads have access to such queue of queues? The Clarion language provides a means to make queue access thread safe using a single synchronization object (as discussed in previous articles). There is, however, a more efficient way to solve this problem when dealing with a queue of queues (or other objects) in a multithreaded environment, as I will show in this article.

Data Declarations

I'll start by declaring two queues – a queue holder (a queue which will store another queue within its record structure) and an inner queue (a queue which will be stored in the record(s) of the queue holder). In use, there will be one instance of the outer queue holding multiple instances of the inner queue.

I'll define the outer queue globally, keeping in mind that multiple threads/procedures might be accessing this queue. At design time I do not know how many records the global queue holder will have, since all the records will be added to it at run time, and their number can vary. I will declare the inner queue with the `TYPE` attribute – this way it will be just a type definition, and no memory for the structure itself will be allocated just yet. At runtime, whenever there is a need for another global queue holder record and, respectively, a "to be stored" inner queue – I can dynamically create a new instance of inner queue using its definition, and Clarion then will dynamically allocate memory for the new inner queue instance.

Here is the inner queue declaration, followed by the queue which stores a reference of the inner queue's type.

```
GQ Queue, Pre(GQ), Type
QueueId2 String(5)
F1 String(10)
F2 Long
```

```

end
GQHolder Queue, PRE(GQH)
QueueID String(1)
QueueRef &GQ
END

```

Inner Queue Creation and access

The following two lines will create an instance of the inner queue and save it in the global queue holder.

```

GQHolder.QueueRef &=new(GQ)
add(GQHolder, GQHolder.QueueId)

```

The creation of a new inner queue (and queue holder record) can be done at any logically acceptable place in the program. If your program is multithreaded, you will need to use one of the synchronization objects provided by Clarion, which I will talk about later.

The statements above allocate memory on the heap for the new instance of the inner queue. Note that I am saving the reference to the inner queue in the record of the global queue holder (`GQHolder`). From now on I can treat `GQHolder.QueueRef` as a queue of `GQ` type. Records of inner queue can be added, deleted and modified:

```

GQHolder.QueueRef.QueueId2='some value for id'
GQHolder.QueueRef.F1='some value'
GQHolder.QueueRef.F2=random(1,10)
add(GQHolder.QueueRef, GQHolder.QueueRef.QueueId2)
GQHolder.QueueRef.QueueId2='some value for id'
Get(GQHolder.QueueRef, GQHolder.QueueRef.QueueId2)
Delete(GQHolder.QueueRef)

```

If needed, you can free the inner queue and nullify (`&=NULL`) its reference, but keep in mind that if you are not deleting the `GQHolder` record and you reuse the reference, you must allocate memory for the inner queue calling `NEW` again. You *can* reuse the reference to the inner queue after you freed it – in this case the only logical operation which will not result in an error is `ADD()`.

When deleting a `GQHolder` record, which stores the reference to inner queue, you must free the inner queue before you delete the `GQHolder` record. Otherwise, only the reference to the inner queue will get deleted, the inner queue will remain on the heap, and you will end up with a memory leak. Also note that a call to `FREE()` does not clear the record buffer; you must call `CLEAR()` to clear it.

UsingThreads.

Queues are not thread safe, so if a program requires that multiple threads access any queue, a synchronization object must be used. There are several synchronization objects provided by Clarion runtime; I will use `ICriticalSection` for this example. The `ICriticalSection` object will ensure that only one thread

can access and modify the queue at one time.

First, create a global reference of type `ICriticalSection`:

```
SynchObjRef &ICriticalSection
```

Then create a new object calling the `NewCriticalSection()` library function. Note that `NEW()` is not supported for creation of `ICriticalSection` objects:

```
SynchObjRef &=NewCriticalSection()
```

Then, whenever accessing the queue, wrap queue access into wait and release block:

```
SynchObjRef.Wait()
<queue access code goes here>
SynchObjRef.Release()
```

After a thread successfully executes a call to `Wait()` on the `SynchObjRef`, none of the other threads will be able to execute a successful wait on the same object, thus keeping queue access code safe.

This technique works perfectly when locking access to a single global queue. It also will work on the `GQHolder` and "nested" `GQueue(s)`. But now let me expand the scenario. Imagine that there are worker threads, which have to access one of the inner queues (without a predetermination as to which one – the decision on which queue to access is made at runtime, and can be changed), in order to modify or delete queue contents, or to recreate the queue. Naturally, you can wrap the access to the inner queue into the critical section as illustrated above, and you will be sure that only one thread will be accessing the `GQHolder` and then inner `GQueue` at a time. The code will look something like this:

```
SynchObjRef.Wait()
<access GQHolder to get the reference to inner queue>
<access inner queue - modify, delete/recreate>
SynchObjRef.Release()
```

Even though the above will work, and provide safe access to the data structure, it is not the best way to do it, and here is why. Since a particular worker thread will be accessing the `GQHolder` only to obtain the reference to a specific inner `GQueue`, it will not be very efficient to prevent access to `GQHolder` and all the other inner queues.

Inner queue locking

To improve the efficiency, lock `GQHolder` only to obtain the reference to the inner `GQueue` of interest. Then lock the inner `GQueue` of interest for data manipulation. This will make `GQHolder` and other inner `GQueues` available to other threads, while this particular inner queue remains locked. You can do this by storing a synchronization object *together with the inner queue*. Change the `GQHolder` declaration as follows:

```
GQHolder Queue, PRE(GQH)
```

```

QueueID String(1)
QueueRef &GQ
qSynchObjRef &ICriticalSection
END

```

Now for every inner GQueue, GQHolder will also store a reference to this inner queue's own ICriticalSection object, on which the thread will call wait() and release() methods in order to provide thread safe access to this inner GQueue *without* blocking access to other inner queues or GQHolder. Don't forget to initialize the qSynchObjRef object to make it useable. You can do this when creating a new GQHolder record. Also, make sure you destroy the object before deleting the GQHolder record to avoid memory leaks.

In order to access the inner GQueue from the worker thread procedure, declare two local variables, which will store the reference to the inner GQueue and the reference to this inner GQueue's ICriticalSection object. Call wait() on the global SynchObjRef to lock GQHolder in order to safely obtain the record of interest, and save references to the inner queue and ICriticalSection object in the local variables. Immediately after doing so, and before you update the inner queue, release SynchObjRef – this will allow other threads to access GQHolder, if needed:

```

SynchObjRef.wait() !lock main queue
clear(GQHolder)
GQHolder.QueueId=j !j is a valid GQHolder.QueueId value
get(GQHolder, GQHolder.QueueId)
assert(~errorcode(), 'error=' & errorcode() & ' after get from main queue')
lQueueRef&=GQHolder.QueueRef
lSynchObjRef&=GQHolder.SynchObjRef
SynchObjRef.release() !release main queue

```

Now you can proceed with manipulating the inner queue's data, using the local reference to the queue. Remember to lock the inner queue's synchronization object (using the local reference to that object) before accessing the queue's data. Also do not forget to call lSynchObjRef.release() when finished accessing/modifying the data.

Here are a few examples:

```

!clear all records...
lSynchObjRef.wait() !lock inner queue
clear(lQueueRef)
free(lQueueRef)
lSynchObjRef.release() !unlock inner queue
!add new records...
k=random(1,10)
lSynchObjRef.wait() !lock inner queue
loop l=1 to k by 1
  clear(lQueueRef)
  lQueueRef.QueueId2=random(1,5)&'-'&j
  lQueueRef.F1='test' & j&'-'&random(1,5)
  lQueueRef.F2=random(1,100)
  add(lQueueRef, lQueueRef.QueueId2)

```

```
end  
lSynchObjRef.release() !unlock inner queue
```

Writing the worker thread code this way allows safe and efficient access to GQHolder and inner GQueue(s).

In this article I have shown how to create and maintain a thread-safe queue of queues, using reference variables and synchronization objects. A test program accompanying this article illustrates the concepts I have described.

[Download the source](#)

[Svetlana Zusman](#) was born 1976 in Ukraine, and moved to the United States in 1998. She graduated from the Ohio State University with a degree in Computer Information Science, and is now working for a small software company in Columbus, Ohio. Svetlana started learning Clarion in 2003.

Reader Comments

[Add a comment](#)

Clarion Magazine

Brazilian ConDev - Notes and Photos

Published 2005-06-13

...and that's the weather, thanks Bob, looks like more snow in Montreal again, but hey, what else is new?

In other news, it's being reported that the Clarionites of South America are gathering down in Iguassu Falls (no I don't know where that is either) and we're going now live (well as live as a printed format allows), to our man on the spot, Muito Louco.

<static> Good evening Dave, yes, I'm here in Iguassu Falls. This is a small town on the southern border of Brazil, where Brazil, Paraguay and Argentina share a common border. The main reason for the town being here are the spectacular waterfalls.

Anyway, back to business. The VI Condev Mercosul Gescla is, as the name suggests, the sixth annual gathering of South American clarionites. Sponsored by the local distributors UniSolutions (Argentina) and Clarion Software Brasil, it is nevertheless completely organized and run by local Clarion users.

The event kicked off with welcome speeches on Thursday night, followed by technical sessions on Multi-DLL programming and Runtime linking.

Friday started promptly with workshops at 8am, and technical sessions started at 11:30. Bruce Johnson from CapeSoft entertained during the first session with a session explaining the three new file drivers. For diplomatic reasons Robert Zaunere was unable to attend, but in a live-linkup after lunch a video conference allowed Bob to present material on the future of Clarion, as well as an exciting look at the new Clarion 7 IDE. This was the first viewing of the IDE outside Soft Velocity itself, and as such attracted a lot of attention, and comment. One unconfirmed source described the IDE as "very slick, very like MS Visual Studio. Certainly a massive step forward for Clarion programmers."

Unlike Devcons run by their lazy counterparts north of the Gulf of Mexico, sessions continued late into the night, with the dinner break being announced (finally) at 10:30 pm.

This is Muito Louco for ClarionMag, in Iguassu Falls, Brazil. Back to you Dave...



A small section of Iguassu falls



**Marco Antonio Machado on
runtime DLL loading**





Talking about the three drivers



Welcome session



**Sergio Baratoja (left) and Adrian (right)
distributers for Brazil and Argentina respectively**



**Jhony Stein presenting an
alternative report writer**



Juan Domingo Herrera with Adriana and Rebecca (local conference organisers)



Luis and the Clarion Basics workshop





Our reporter on the way to touch-typing class...

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](#)

- [» "lazy counterparts north of the Gulf of Mexico", in a pigs...](#)
- [» I still believe that took me a picture of me with the two...](#)

Clarion Magazine

Brazilian ConDev Report

by **Fernando Cerini**

Published 2005-06-13

CONDEV is an event organized every year since 2001 by GESCLA, the Brazilian Users Group. GESCLA is a very "democratic" group of very nice people.

All the events carried out by GESCLA are events organized by users of Clarion and for the Clarion community. The objective of each organizing commission is to improve the event year by year, taking advantage of the experiences of the previous organizers.

Evolution Consulting has had the good fortune to be present and to participate in the two last events (Porto Alegre 2004 and Foz do Iguazu 2005). This year's conference featured Bruce Johnson from Capesoft, who gave two excellent presentations: Using the ErrorClass for Error Handling in ABC, and A Tale of Three Knights, about the new Softvelocity drivers.

Unfortunately, Robert Zauere, president of SoftVelocity, could not attend due to problems with the visa required to enter Brazil, so he and Diego Borojovich gave their presentations by videoconference. Attendees at the Conference were the first outsiders to see the new Clarion IDE. Robert presented an overview in English, and Diego discussed the new language features of Clarion.NET, in Spanish. These presentations are available for download from Clarion Magazine.

This is the list of Workshops and presentations:

- Clarion Workshop - Luis Dentati
- Java and Jaguar Workshop - Juan Herrera and Matias Flores.
- PostgreSQL Workshop - Rogerio Basseti.

- Evaluate and Dynamic expressions - Alejandro Dellacanonica
- Atomic Clarion - Gustavo Pinsard. About the power of the language and how to use Clarion without templates.
- Clarion and XML - Ricardo Lopez Celani.
- Web & HTML - Ricardo Fantin.
- Report Manager - Jhony (Stein) Pereira. One of my favorites, more info [here](#).
- Multi driver Apps - Natalia Perazza. About how to use the same app to connect to different database servers.
- Developing big applications - Alejandro Dellacanonica.

Detailed information will be available at <http://www.gescla.com.br> as soon as possible.

Congratulations to the members of GESCLA for this great event, and thank you so much for inviting us and to make us feel like at home. We made a lot of new friends.

Next year there will be two CONDEVs: one in Bonito, Brazil, and the other in Punta del Este, Uruguay. Both are very nice places. Evolution Consulting is already the sponsor, and will help in the coordination and organization, especially in Punta del Este with Fernando Rey and our friends from Macrosoft.

This is my humble report, but if you really want to know what a CONDEV is, you must come here and see it for yourself!

[Fernando Cerini](#) is a Software Engineer with more than 10 years of experience in Clarion. Until 2003, he was in charge of research, support and training for the local Clarion distributor in Argentina. Now he works for [Evolution Consulting.com](#), the Clarion International Distributor for Latin America, which has carried out numerous courses and conferences in the region.

Reader Comments

[Add a comment](#)

Clarion Magazine

Putting Clarion 6 Under Version Control With TortoiseSVN

by David Harms

Published 2005-06-08

Some years ago I regularly used version control to manage my software development, with tools such as StarTeam and MS Source Safe. But I somehow got out of that habit, relying instead on regular backups. Then I received [several articles](#) for Clarion Magazine on CVS and WinCVS, which piqued my interest. Mark Geisinger's [article](#) last month on Subversion prompted me download and install Subversion, and the TortoiseSVN Windows client for Subversion. TortoiseSVN makes version control quite painless.

Although version control is obviously a useful technique in managing your own code base, it can also be convenient to put Clarion itself under version control. Although Clarion does provide a way of uninstalling patches, for instance, it's sometimes handy to be able to quickly produce a particular version of Clarion, or to compare changes between two versions. In this article, I'll show you how you can use TortoiseSVN and Subversion to manage Clarion upgrades using a local repository.

Getting started

I suggest you begin by reading [Mark's article](#). Although I'll cover some of the basics, Mark goes into more detail. Also, Mark used a Subversion repository running on a Linux box; I'll describe a local repository, running on my Clarion development machine. If you're not sharing code with other developers, a local repository works just fine.

I'll also be describing how I put three versions of Clarion (6, 6.1, and 6.2) and associated patches under version control. If you've already upgraded to 6.2 and you're not much concerned with previous versions, that's no problem. Just go through the import/checkout process as described, and you'll be in a position to commit any future patches to 6.2.

Installation

Subversion and Tortoise are very easy to install. Just download and run the installers:

- [Download](#) and install Subversion
- [Download](#) and install Tortoise SVN

I did run into a problem with TortoiseSVN when I first installed it – right-clicking on most Explorer folders caused Explorer to crash. This was a non-trivial problem, but I eventually figured out that the culprit was PowerDesk 4.0. I uninstalled that product, and TortoiseSVN worked fine. Mark Geisinger has told me that there are no issues between TortoiseSVN and the 6.0 release of PowerDesk.

Creating a local repository

Subversion needs a place to store all those versions of whatever files you want to track. This is called a repository, and as I indicated earlier, mine is a local, non-shared repository, which exists on my Clarion development machine.

To create a local repository, make a new folder. Mine is C:\SVNRepository. Next, right-click on that folder and choose TortoiseSVN|Create Repository here. You will be asked whether to create a Berkeley DB or native file system repository – I chose the default Berkeley DB only because it's the default, and has been around longer. BDB won't work for a shared repository, however, as it is not suitable for use over network filesystems. If you wish to store your repository on a share on your LAN, you should create an FSFS database.

Using TortoiseSVN

TortoiseSVN is available through the Windows Explorer, or any explorer-like tool. You initiate tasks from the context menu (Figure 1), which you get by right-clicking on a file or folder in Explorer.

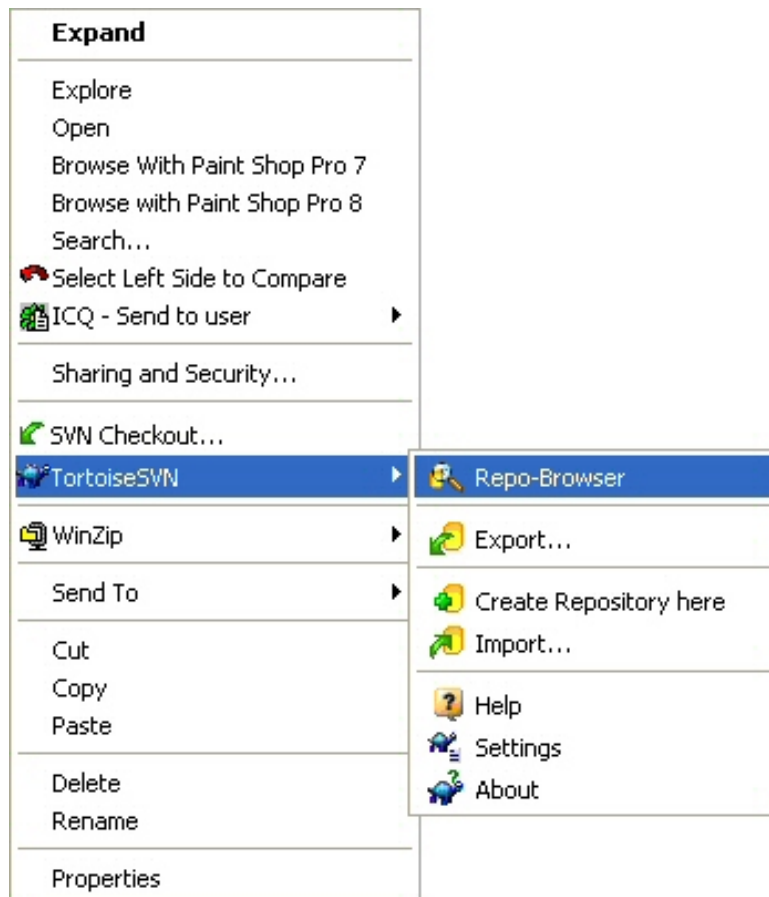
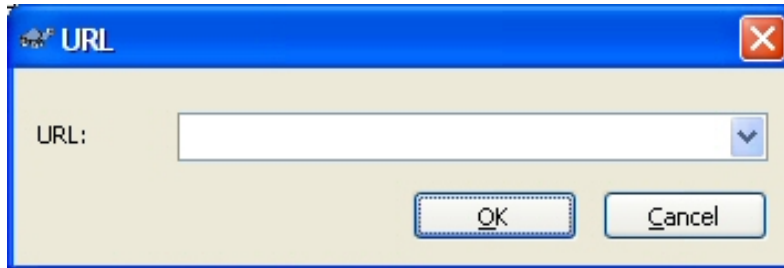


Figure 1. The Tortoise context menu

Selecting a repository

You can have any number of repositories on your local machine, or available on other servers. Your first task, then, is to tell TortoiseSVN which repository you want to use. Right-click on the repository folder (actually any folder, or even the Desktop, will do for this command) and chose TortoiseSVN|Repo-browser. The first time you use the browser you will need to type in the path to and name of the repository – after that it will remember this information. See Figure 2.

**Figure 2. Choosing a repository**

For a repository created in the folder C:\SVNRepository, you would type:

```
file:///C:/SVNRepository
```

Now it's time to set up the repository. There are two things to consider here: the recommended basic directory structure, and the kinds of projects you will want to keep under version control.

As Mark indicated in his article, the basic recommended structure is as follows:

```
/trunk
/branches
/tags
```

The trunk directory will contain whatever you consider your "main" version of the project. If, say, you decide to release a new version of your program, you will create a "tag", which is basically a marker that lets you go back and retrieve all of the files in the state they were in when the tag was created. In my example of putting Clarion 6 itself under version control, all the Clarion 6 subdirectories will end up under /trunk, and the various versions of Clarion 6 will each have their own directory tree under /tag, such as /tag/Clarion6Gold or /tagClarion6.1Gold..

You probably won't need branches for Clarion itself. On the other hand, if for instance you find you must create a special version of one of your own projects for just one customer, you would create a branch, not a tag. The general idea is that tags are a snapshot of your project at a particular time, and branches are ways for you to maintain multiple versions of your program at once.

Subversion doesn't actually keep multiple copies of your projects when you do branches and tags. Instead, it uses what it calls a "cheap copy" which is simply a pointer back to the data stored in the trunk. If you make

modifications to a branch, only the changes are stored; for everything else Subversion just uses the trunk data.

It's important to differentiate between projects within the repository, of course, so you need more than just the trunk/branches/tags structure. The typical options are:

```
/trunk/projectA  
/trunk/projectB  
/branches/projectA  
/branches/projectB  
/tags/projectA  
/tags/projectB
```

or

```
/projectA/trunk  
/projectA/branches  
/projectA/tags  
/projectB/trunk  
/projectB/branches  
/projectB/tags
```

The first approach makes more sense if you have very closely-related projects, the second if you have very different projects. In the case of C6, I've used the following structure:

```
/Clarion6/trunk  
/Clarion6/branches  
/Clarion6/tags
```

To create these directories, use the Repo-browser, right-click (in my case) on SVNRepository, and choose Create Folder. Type Clarion6. Then do the same with this folder, creating the trunk, branches, and tags folders. Your directory tree should now look like Figure 3.

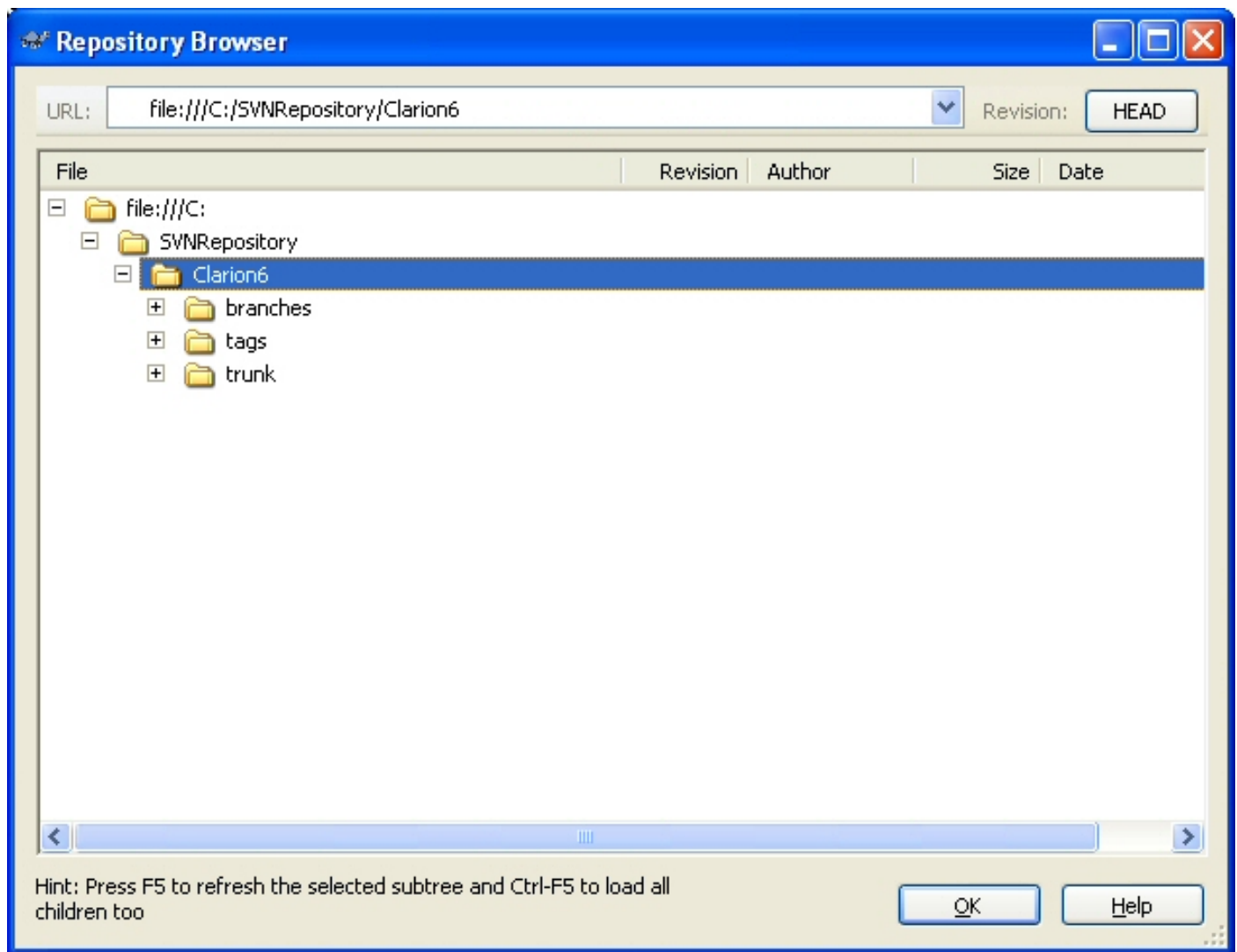


Figure 3. Creating the directory structure

At this point, you are almost ready to do the initial import of Clarion 6 to the repository. But first, clean out (or exclude) any unnecessary files, as Mark describes in his article.

Importing the initial version

I decided to go all the way back to Clarion 6.0, so I'd have an end-to-end history of Clarion versions. You might want to pick up the process as of 6.1.

As Mark noted in his article, you should always have backups of the directories you're putting under version control. Although that sounds overly redundant, it's simply a precaution against making a rookie mistake. And although version control itself acts as a kind of backup, don't forget that even if you're using version control perfectly, you still need, at a minimum, to backup the repository!

It's very important to note that when you first put a project under version control, you cannot just magically mark it as being under Subversion's management. You have to suck everything up into Subversion, and then spit it out again, in two distinct steps. Perhaps at some future date this process will be made more seamless, but this is how it works now.

After making a backup copy of your Clarion directory, right-click on that directory and choose Import. The dialog shown in Figure 4 appears.

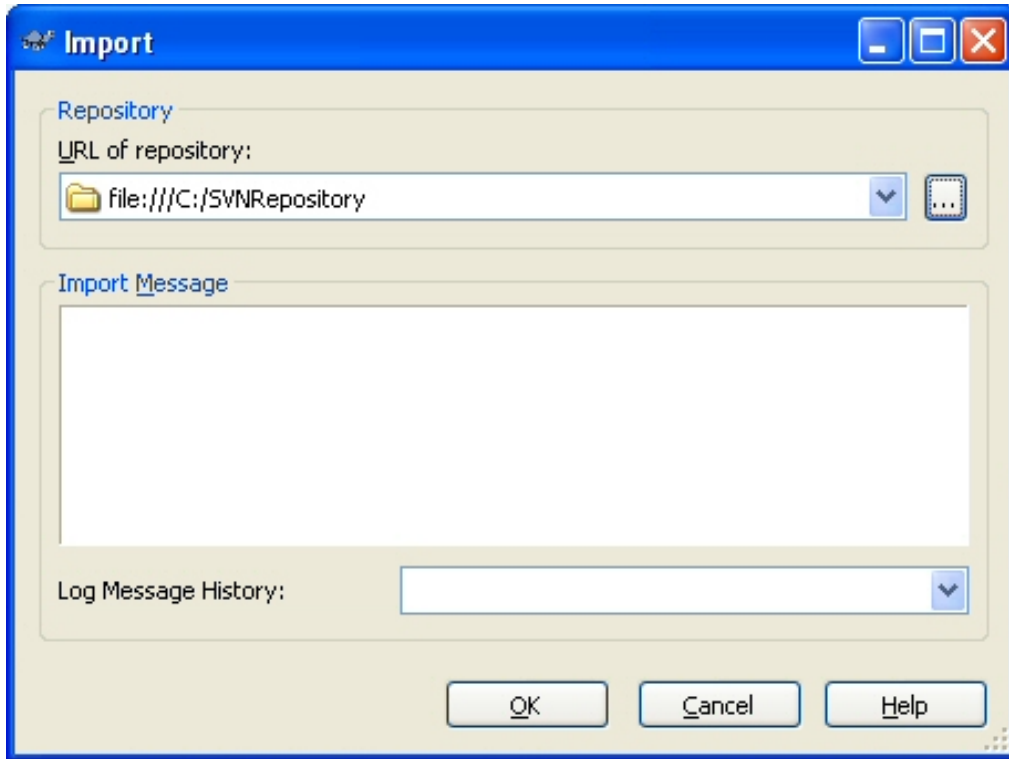


Figure 4. The Import dialog

Note the [...] lookup button to the right of the URL droplist. Click on that to bring up the repository browser, as shown in Figure 5.

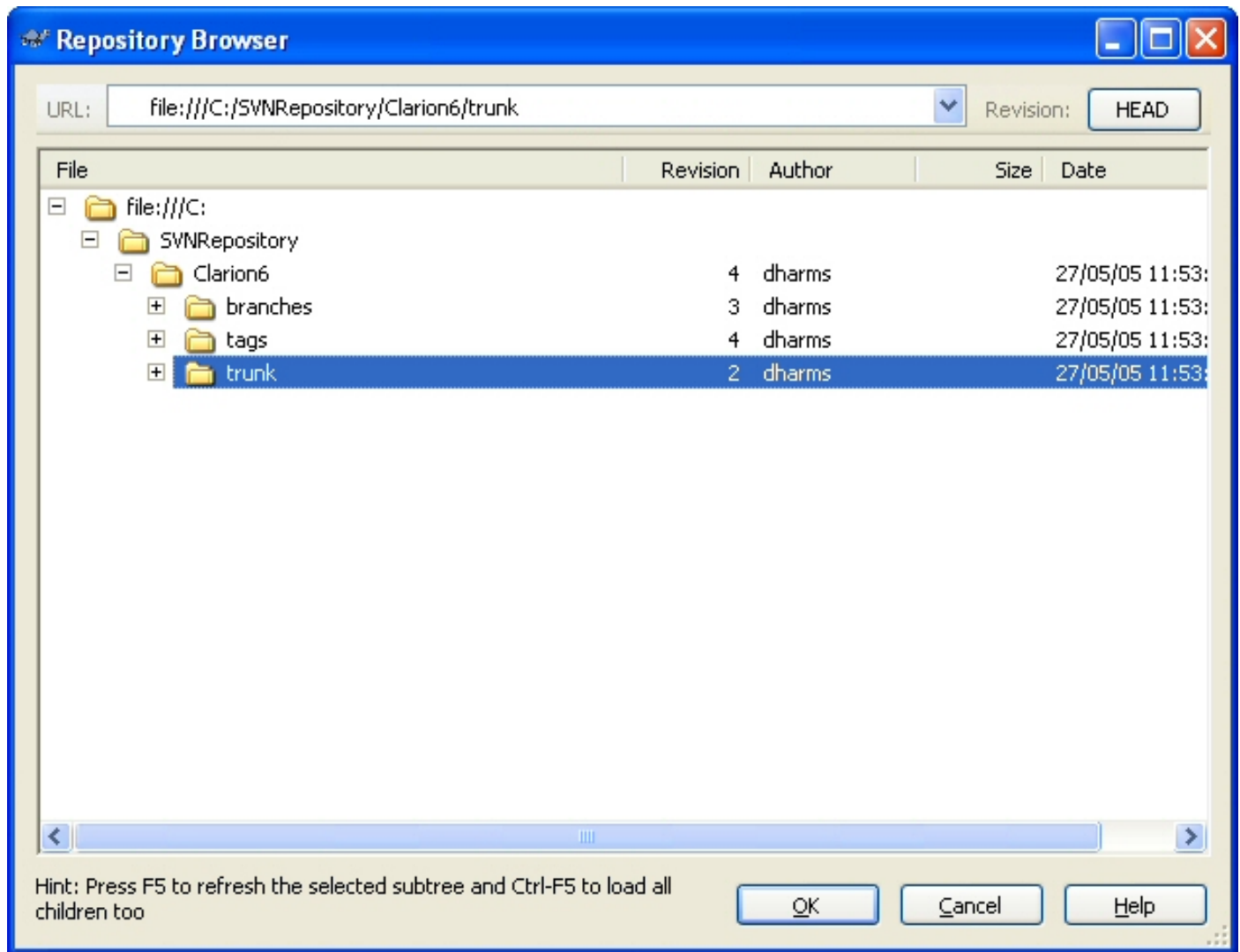


Figure 5. Choosing the trunk directory with the repository browser

Be sure to select the Clarion6/trunk directory (or equivalent) as the "import to" location.

Figure 6 shows the import window. Note that I've added a descriptive comment to the effect that this isn't a completely pristine C6 installation as there are some other products installed as well. It is, however, important to include third party products and anything else necessary to maintain a particular version of your applications. (In fact, at the same time as you create a tag for a major release of your software, you should probably create a tag of the corresponding Clarion install as well, so you can always go back to the complete build environment at some later date, if necessary).

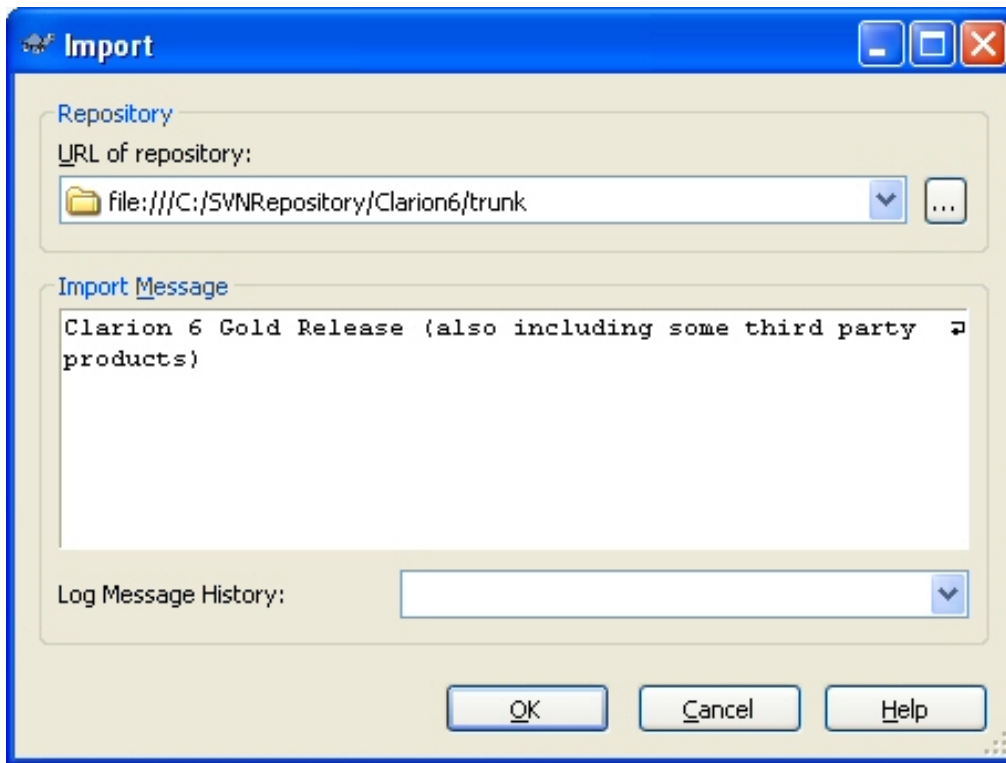


Figure 6. The import window

Feel free to go back and make sure that you *did* in fact right-click on the correct directory.

If you're all set, then click OK to begin importing. The import log window (Figure 7) will show the files as they are imported.

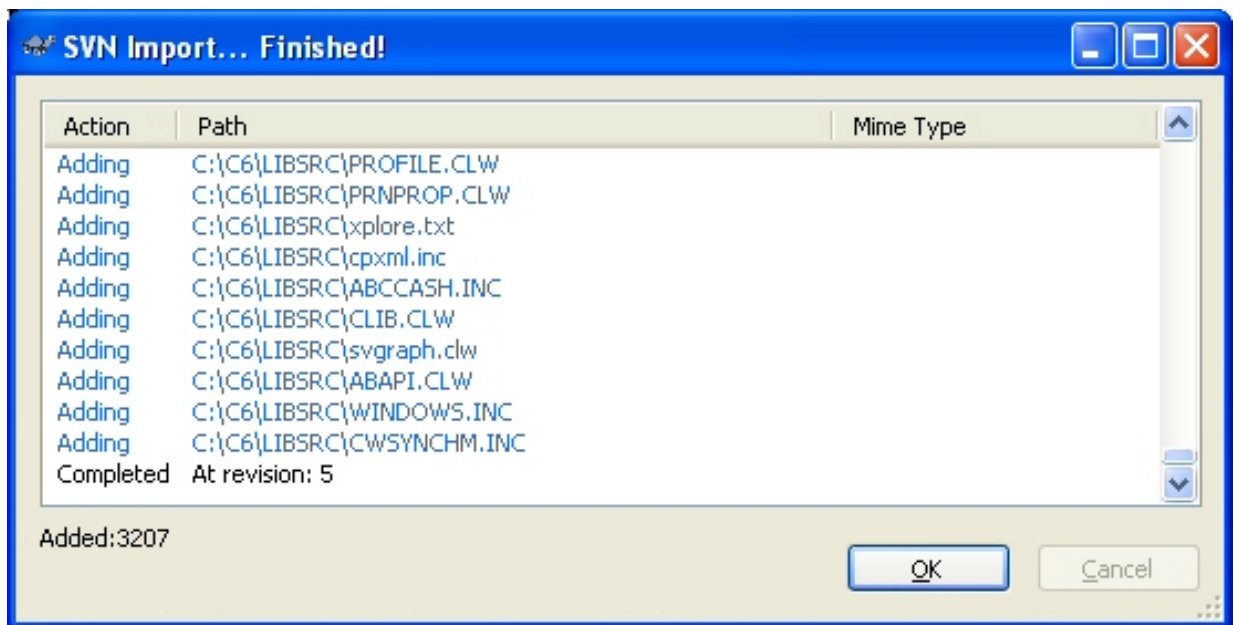


Figure 7. The import log window

Now go back and delete (did you make a backup?) or rename your original Clarion directory. Create a new directory with the same name as the original. Now right-click on that directory, but instead of going to the TortoiseSVN subdirectory, choose the SVN Checkout command. The dialog shown in Figure 8 appears.

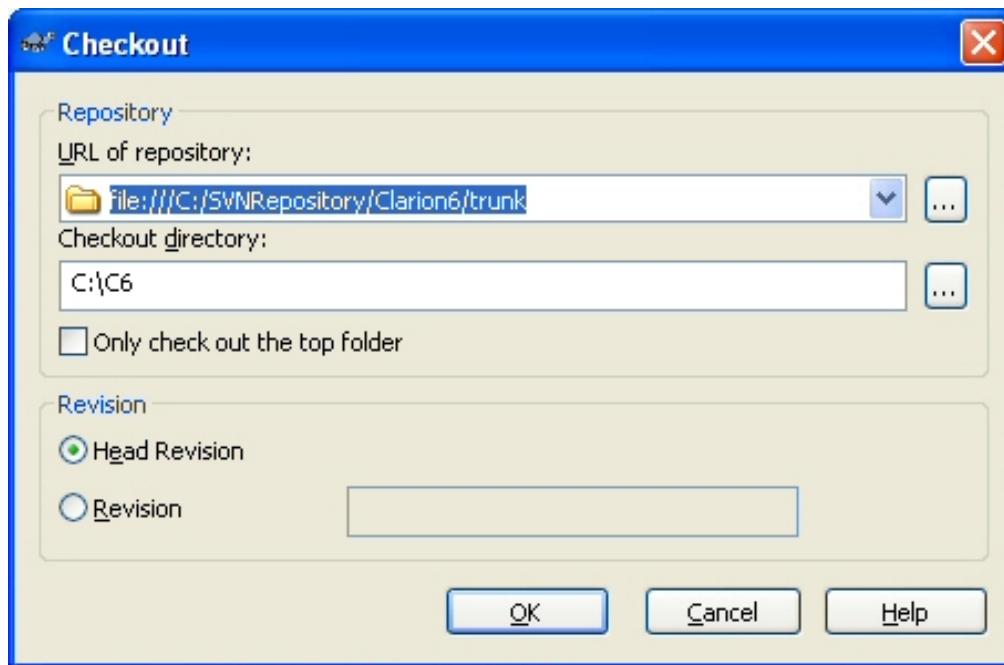


Figure 8. The checkout dialog

Choose your just-created repository, and check out a head revision as indicated in Figure 8. "Head" simply means the most recently-committed version.

Next, use the context menu to choose Branches/Tags. Create a tag for the just-committed C6 Gold release as shown in Figure 9.

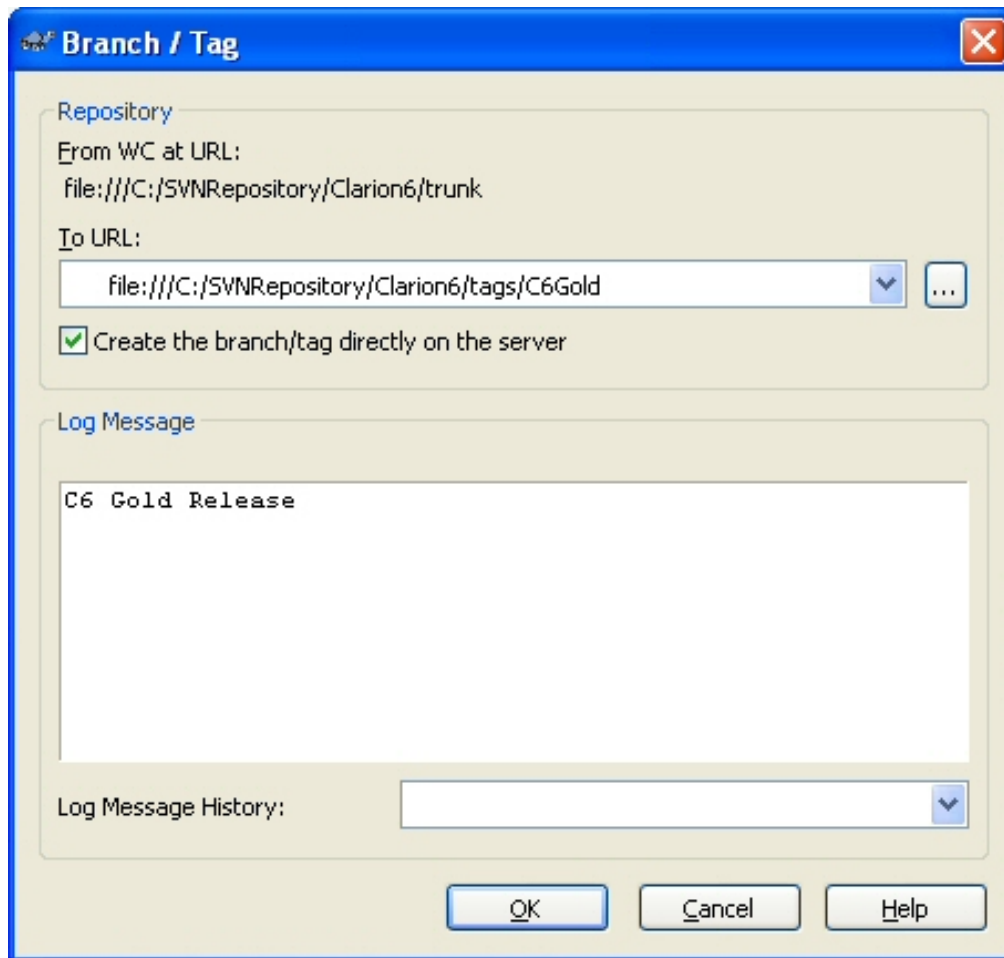


Figure 9. Creating a tag

You may need to refresh the directory list (F5) to see the branches directory.

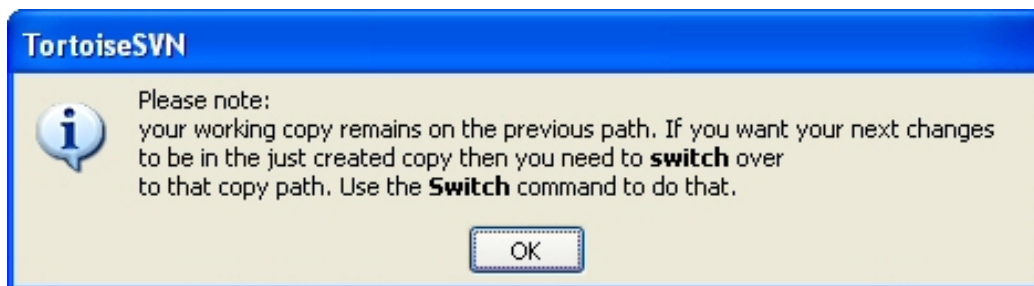


Figure 10. The Switch warning

The switch warning appears after the tag has been created. The point of this warning is that although you've created a tag (which is really just a pointer to a particular revision), Tortoise still considers your working copy to be the trunk copy. That's not an issue in this case since the tag is there simply for future reference, should I want at some time to revert to the revision indicated by the tag. (Actually I could dispense with the tags and keep track of the revision numbers myself, but the tag feature is convenient and inexpensive and produces a nice graph.)

A full discussion of switching projects is beyond the scope of this article, but basically you have two options for updating your working copy to any particular revision stored in the repository. These are Update and Switch, which are similar, except that Update moves everything out of the repository, and Switch just moves the changes, and so is generally a lot faster. But both will merge any changes from the repository into your working

copy, so if you're wanting to go back to a previous release of Clarion, you're probably better off checking out the appropriate revision to a new directory.

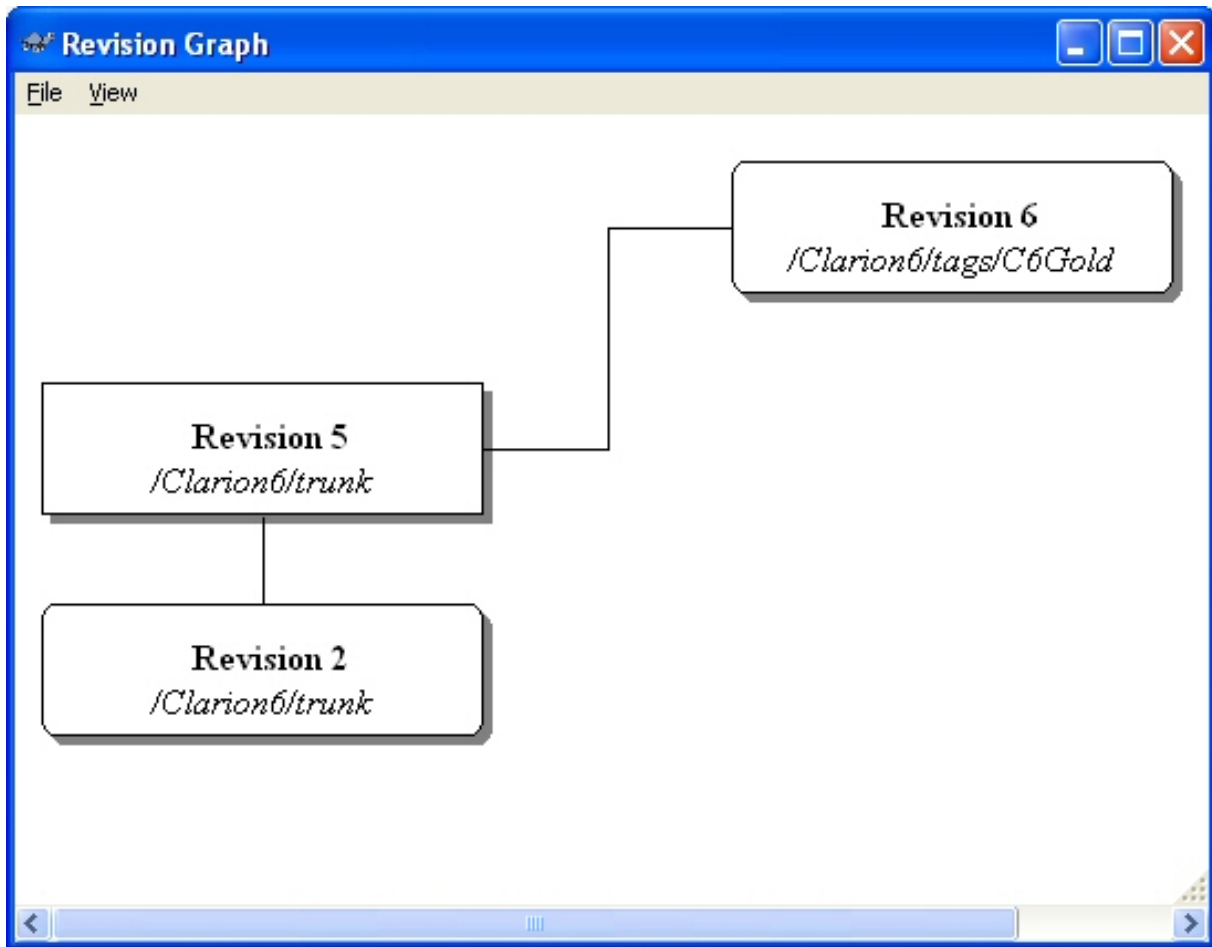


Figure 11. The revision graph

Okay, now it's time to upgrade to C6.1. Since I'm on a standalone repository I don't have to worry about anyone else's changes, and I can just go ahead and apply the patch.

After the patch runs, I want to find out what's changed, and particularly what's new. This is because Tortoise won't automatically commit new files to the project.

I can either look through all the directories to see what's new, or I can use the Check for Modifications command, which gives me a list of all files. Figure 12 shows a portion of the log.

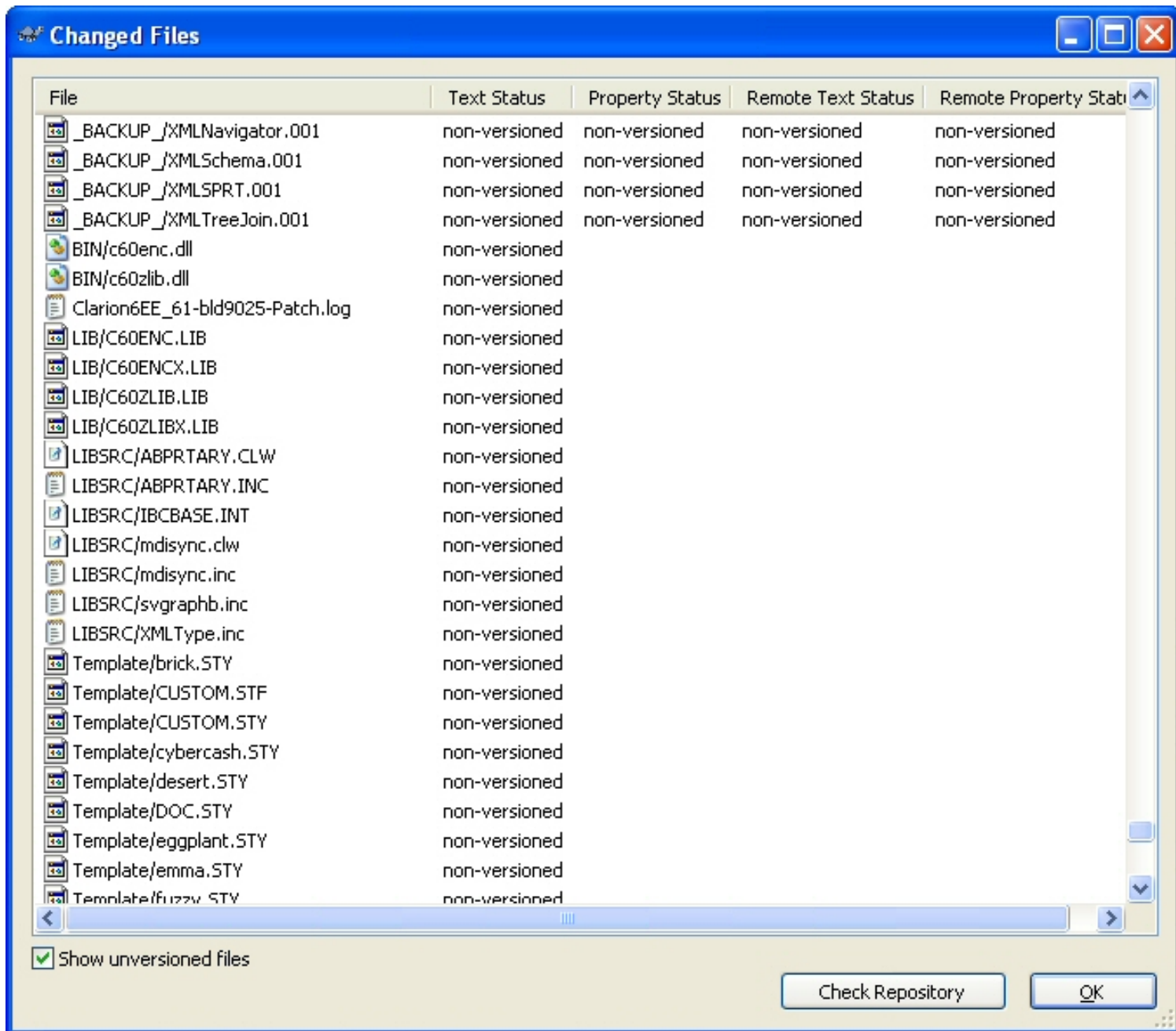


Figure 12. Checking for modifications

The backup files may or may not be important, depending on whether I want to rely on Tortoise to manage a regression, or whether I still want the option of running the Clarion uninstall program. But clearly new DLLs, templates, and source files need to be added. You can do this when you commit, or you can first call the Add command, and then do the commit. In either case you need to keep in mind that new files are not automatically added to the repository.

You can select the top level directory and choose the Add command, which will present you with a list of all possible files to add, from which you can select/deselect (Figure 13). If you deselect a directory, all the files in that directory are also deselected, but not vice versa.

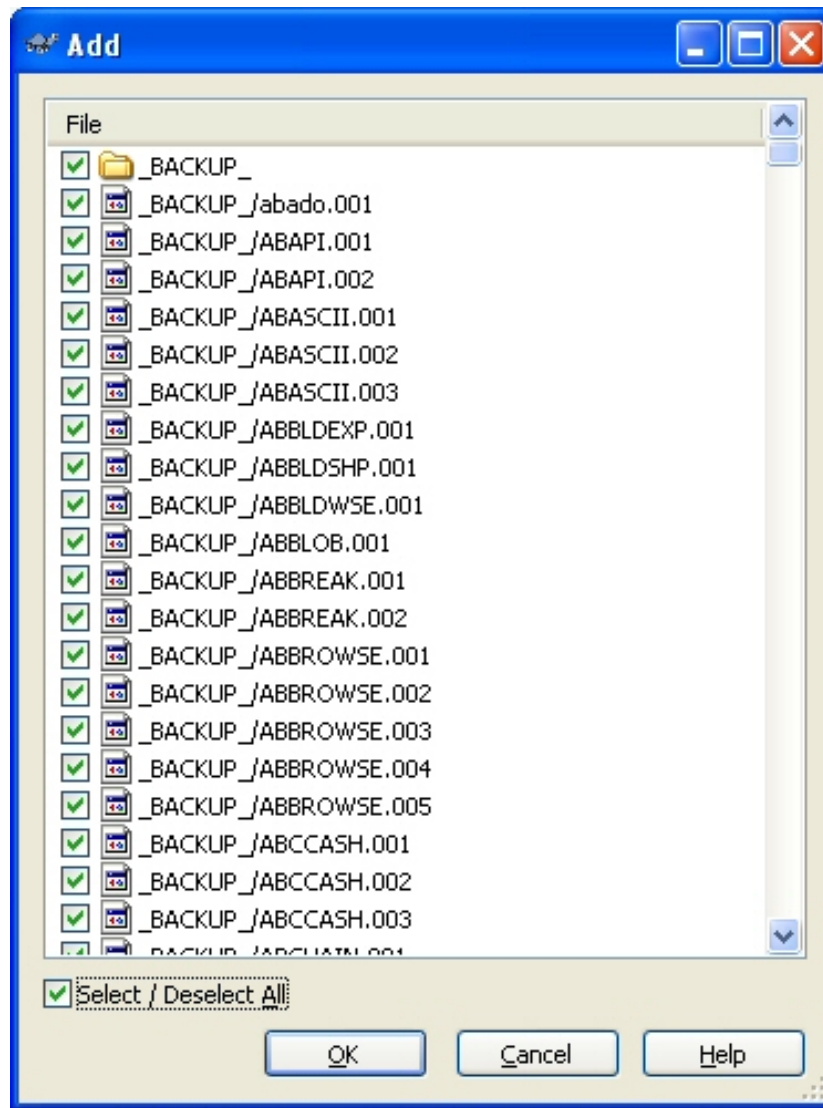


Figure 13. Adding new files

You're not done yet - "adding" only marks the files as "to be added", it doesn't put them in the repository. To do that you need to do a commit. When you do a commit, you will still have the option to add new files, as shown in Figure 14. You will need to manually check the files you want to add.

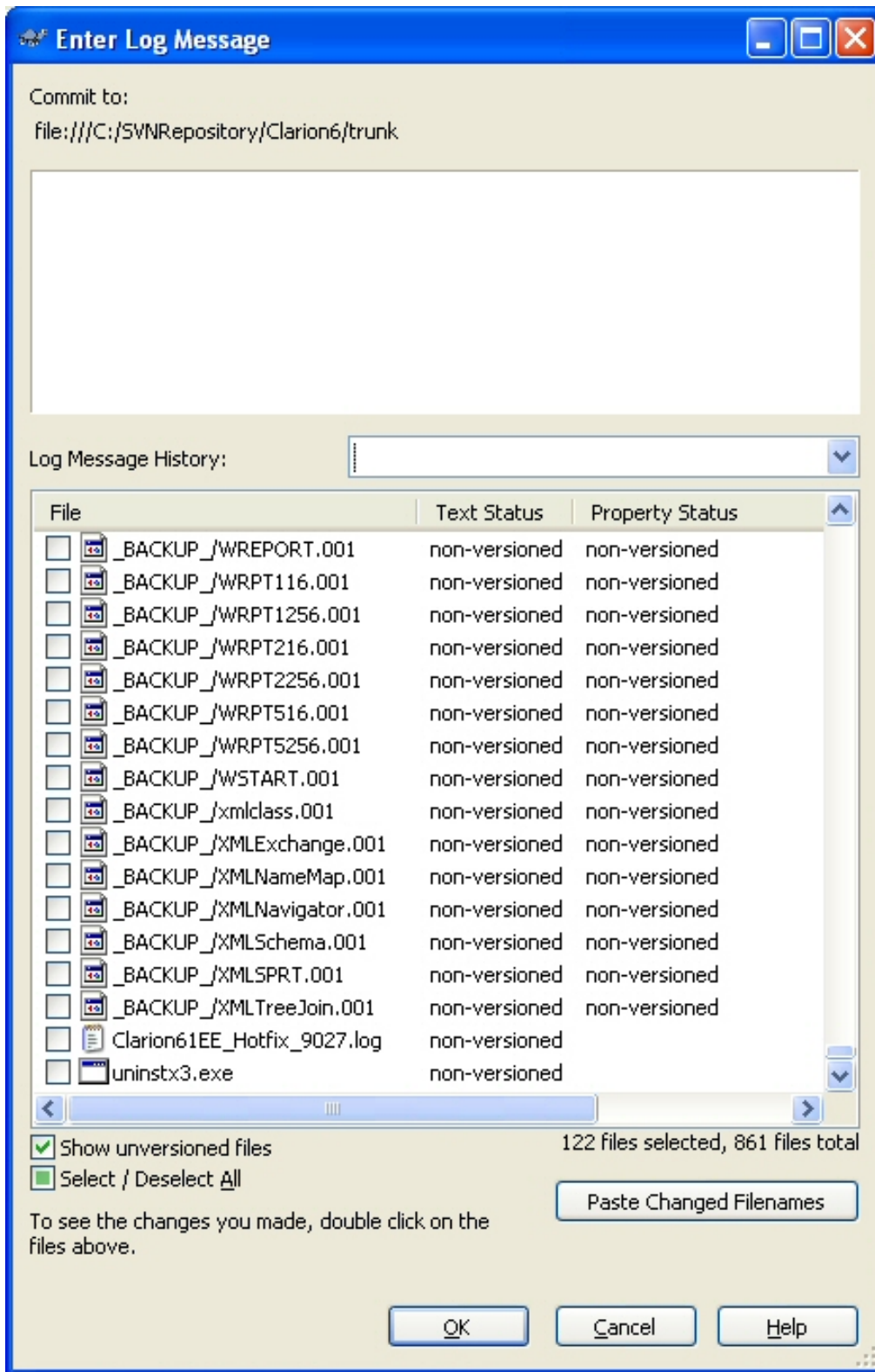


Figure 14. Adding new files at commit time

After you added the files/directories to source control the file appears with a added icon overlay which means you first have to commit your working copy to store those files/directories in your repository. Simply adding a file/directory does not affect the repository.

After upgrading to 6.1, and creating another tag for that release, I applied all of the 6.1 patches up until the last one before 6.2 and committed the changes to the repository. I then created a tag for the 6.1 final release. Next, I

applied the 6.2 patch, committed the changes, and created a tag for that release.

On each commit I made sure to add the new files – there weren't a lot of these in the 6.1 patches, mostly just new uninstall EXEs, but the 6.2 patch introduced a number of new files.

Viewing changes

Viewing differences between your working copy and the repository is easy. To see differences between entire directories, choose TortoiseSVN|Check for Modifications from the Explorer context menu. You can also check changed files individually via the TortoiseSVN|Diff menu option, which appears only when a file has changed and has not yet been committed.

Comparing revisions

I haven't found a really easy way to compare two committed revisions in Tortoise, although you can use Show Log to display the revision list, select two revisions, and generate a unified diff file. I don't consider that a really easy way to see, for instance, what's changed between 6.1 and 6.2. The time to do something like that is before you commit the latest patch or changes. Alternatively, you could export a revision to another directory and then use a tool like Beyond Compare to compare that revision with your working copy, or another exported revision. Like Checkout, Export places a copy of a project, or selected files, on disk for you. Export differs in that the exported files are not under Subversion's care.

Tip: You can quickly export select files or folders from a project directory by *right-dragging* them to a new location. When you drop the files/folders under version control into their new location, TortoiseSVN adds additional export options to the context menu.

Other issues

Using TortoiseSVN to manage Clarion versions is a straightforward task, provided you don't modify any of the shipping templates/source code. Merging changes is beyond the scope of this article, but there is quite a lot of information available in the online help – just search for "Merging". And of course the beauty of version control is that even if you completely forgot that you'd made some change in one of the shipping source files, and that file gets overwritten on an upgrade, as long as you've committed the previous version you can always go back to it, compare it, and migrate any necessary changes.

If you already have a version of TortoiseSVN prior to 1.2 installed, and you're upgrading to 1.2, you'll need to go through a [separate process](#) to make the repository accessible to the 1.2 release. Do this *before* you upgrade as you need to use the older svnadmin command line tool to prepare the repository for the upgrade.

Summary

On the whole I've found TortoiseSVN very easy to use, although when first setting up a project I would personally rather have an application that stayed open, rather than one that required me to use Windows Explorer

context menus every time I wanted to take an action. But for most tasks it's easy and convenient. After too many years away from version control, I'm in the process of getting all my source code safely in the hands of TortoiseSVN.

Resources

Free [Subversion E-Book](#) (doesn't cover TortoiseSVN, but provides good information on how Subversion is handling things behind the scenes, as well as version control basics)

[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 produces the [Planet Clarion](#) podcast, which he co-hosts with Andrew Guidroz II.

Reader Comments

[Add a comment](#)

Clarion Magazine

Planet Clarion Transcript: Clarion 6.2 Released

Published 2005-06-03

Dave Harms: This is Planet Clarion for Wednesday, May 25, 2005. I'm Dave Harms.

Andrew Guidroz: And I'm Andrew Guidroz.

Dave Harms: Good day.

Andrew Guidroz: Good day.

Dave Harms: So on today's show we're gonna be talking about the big news out of SoftVelocity, the release of Clarion 6.2, which people have been waiting for, for quite a long time.

Andrew Guidroz: Yeah.

Dave Harms: Yeah, and there's also gonna be a bit of a surprise for some people because 6.2 is not a free release.

Andrew Guidroz: Right, it's not free.

Dave Harms: Unless you bought, what unless you bought Clarion in the last six months I think? Is that right?

Andrew Guidroz: If you bought it, I believe, since November 1st.

Dave Harms: Since November 1st.

Andrew Guidroz: Of 2004.

Dave Harms: That's right, yeah. If you purchased a Clarion 6 Professional Edition or Clarion 6 Enterprise Edition upgrade, or a new license, after November 1st, 2004, you are entitled to a free upgrade. For everyone else, there's a minimum 99 dollar...

Andrew Guidroz: 99 bucks, right.

Dave Harms: 99 dollar upgrade charge and there are some subscription programs, CSPs available now, which we're gonna talk about in a little bit as well. So there are a couple of different ways you can get that. And we're gonna talk about the whole business of why we're paying for 6.2 and...

Andrew Guidroz: You got it.

Dave Harms: ...what that all means, but maybe we should talk a little bit about 6.2 first, so...

Andrew Guidroz: Right.

Dave Harms: ...let's do it.

Andrew Guidroz: Well, the first thing to jump at with 6.2 is directly to the new features. I mean, the idea that SoftVelocity evidently had in this thing is that they're a business and needed some income, and as part of this upgrade, they've added some new functionality.

Dave Harms: Yeah, it's not just a bunch of bug fixes. There are actually some substantial new items in there. And I gotta say one of my favorites... maybe we can just go through a few of those right now.

Andrew Guidroz: Go ahead.

Dave Harms: There are some core template stuff which I like but actually the first thing I wanna talk about is the improvements to the SQL drivers and there.

Andrew Guidroz: Ooooh.

Dave Harms: Yeah, there's this new thing called TurboSQL and I don't, you do a bit of SQL stuff, right Andrew?

Andrew Guidroz: I sure do, yeah, definitely.

Dave Harms: And I know people who do SQL often run into this thing where in order to use, to execute an SQL statement against the back end, you have to be using a table that is defined on the database server, right?

Andrew Guidroz: Right.

Dave Harms: You can't just sort of write up a, you know, create a file definition and with a bunch of arbitrary fields and then do a select statement that returns those fields to that. You actually have to have a matching file. It doesn't have to have, your dictionary definition doesn't have to have all the fields, but it has to have whatever fields are in the dictionary definition also have to exist on the server. In 6.2, that's no longer a requirement. You can actually define a table structure and you do not have to have a matching table structure on the server, so...

Andrew Guidroz: Exactly, so you get rid of all that overhead right away where every time you go to open a table, on your SQL side, what was happening before was there was this great dance going on in the file driver that was making sure the SQL side and your dictionary side matched up.

Dave Harms: Yeah.

Andrew Guidroz: Now, if you turn on the turbo, you're gonna see an instant speed increase, because that little dance of, you know, this is what I think you look like, what do you really look like doesn't have to happen anymore.

Dave Harms: Mm-hmm. Yeah, and I would say to me the biggest thing actually is not even the speed increase, because I think I could live with the same speed that I've seen, but it's the flexibility. Because you can now do calculated fields easily, if you wanna just, you know, you've heard about the great...

Andrew Guidroz: The temp table...

Dave Harms: What is it? Troy Sorzano was the first guy to write this up for Clarion, right?

Andrew Guidroz: Right.

Dave Harms: Uh, what did he call that, the stupid temp table trick.

Andrew Guidroz: Exactly.

Dave Harms: Where you create this table and then you use it basically only to receive the results of an SQL statement so that you can read it in your application, but what you don't need now is you don't need to have those temp tables and you don't have to be hamstrung by whether you remembered to put the right number of fields or the right order of fields or whatever else in that table, so...

Andrew Guidroz: Right, yeah.

Dave Harms: So I think for anybody who's doing any amount of SQL, this is a, I think it's a huge boon. I mean, if you're doing SQL, I would say that this justified the price of the upgrade on its own.

Andrew Guidroz: It's almost like having a file alias and never messing with the current buffer at all.

Dave Harms: Yeah. Or having...

Andrew Guidroz: Not having all the trouble of an alias.

Dave Harms: Yeah, or having a view, only a much easier way to deal with a view instead of having to hand code the view or use some other scheme to create a view, to have the view generate the SQL statement for you. Now you can just create a definition, execute the PROP:SQL and you're off to the races. So it really just, it just brings you much closer to the kind of SQL that you would be doing if you were just sitting at a console somewhere or if you were using, you know, a language that let's you return a record set or something like that.

Andrew Guidroz: Right, it's where you'll be using the PROP:SQL a whole of a lot, <laughs> a whole lot.

Dave Harms: A whole of a lot, that's right. <laughs>

Andrew Guidroz: A whole of a lot.

Dave Harms: <laughs>

Andrew Guidroz: A heck of a lot more.

Dave Harms: Yeah.

Andrew Guidroz: You know, you won't be looping and doing all this client side stuff, you're gonna let the server do the job, which is what SQL does the best anyway. You know, you're gonna say here's my big old select statement, and you can do all this great crunching on the server side and just give me back the numbers.

Dave Harms: Yeah.

Andrew Guidroz: Just that new feature, just that PROP:SQL feature to me for what I'm doing, the headache it removes is a lot of that 99 dollar fee that they're asking for, for 6.2. I mean, that's a whole lot of it for me. I mean that one improvement to the SQL driver. It's important. It's a time-saver.

Dave Harms: Yeah, and we don't want to sound like we're just sort of shills for SoftVelocity here either. I mean, I know there is going to be, there are going to be some objections to paying for the upgrade and we'll get onto those a little later, but I think what we're just trying to say is that there are some actual, some substantial new features in 6.2.

Andrew Guidroz: And there's some value.

Dave Harms: And there is some value. Yeah, there certainly is.

Andrew Guidroz: Now the next feature that's brand new and they've actually put a price on this thing. They're gonna sell this thing outside of 6.2 as well as a new feature, is the new relation tree.

Dave Harms: Yeah, the MultiChild Tree Template.

Andrew Guidroz: Yeah, MultiChild Tree Template, relation tree template.

Dave Harms: Yeah, do you know what the pricing is?

Andrew Guidroz: 69 bucks.

Dave Harms: Oh yeah, it does say there 69 dollars.

Andrew Guidroz: 69 bucks.

Dave Harms: So this one template on its own. But yeah, this is actually kind of a nice thing, and there's a little movie that you can watch to see how the multi-child relation tree template actually works, and I think that's a worthwhile thing for anybody to look at if they're considering it.

Andrew Guidroz: Yeah, it really goes into great detail. I mean, when you read the text on the web site, I don't think it's as profound as when you watch the movie that they've come up to demonstrate it. I mean, literally you've got a queue-loaded relation tree and it loads everything in the memory for the root level, but it'll go up to 29 different levels of children and it'll do those on demand. So you can really get some nice, deep, complex relationships like for instance, if you use Microsoft Outlook, where you've got different file folders and you've got mail and you got calendars and all these different kinds of icons and pictures that show up with each level that pop open and deeper and deeper, um...

Dave Harms: And in fact that's the example that they use is a layout that looks a little bit like sort of an Outlook type tree.

Andrew Guidroz: An e-mail type example, exactly. And I mean, it was, I mean that's the first thing that I thought of. I mean, that's about the most complex tree that I use in any application that I have is Outlook. But that to me is a pretty cool thing.

Dave Harms: Something else listed under core templates, which is something that, it's one of those things that I have, it's bugged me about Clarion for I don't know...

Andrew Guidroz: Forever.

Dave Harms: Yeah, forever, since Clarion for Windows 1.0, which is when you shut down a stock Clarion, when you shut down Windows, if you have a standard Clarion application running, Windows will say "Oh, I can't shut down, you have to shut down this application first." In 6.2, and I don't care how long you say it's been due, finally it's there, I'm happy to see it, there is a shut down with Windows template.

Andrew Guidroz: Hurray!

Dave Harms: Yay! There's also a little bit more to it than that because there's also a, there's a minimize to system tray option and there's also a clean close down template and this is something that if you have a multi-DLL application, will sort of manage the order

of shutting down all the DLLs and I guess unloading them from memory. So that's some stuff that's very welcome.

Andrew Guidroz: It's a non-trivial template. I mean, it's something that if you've gotta shut down with the new threading model and all the different multi-DLLs you've got going on, you know, you see different techniques, someone's always asking "Well, how do I shut down each thread?" Well, you loop through this and you post events and you do this and you do that and someone always says "Well, no, but that doesn't account for this" well now you've got the solution already in the box. It's pretty cool.

Dave Harms: Yeah, and you know, in all fairness to SoftVelocity and TopSpeed before it, Clarion applications are, especially multi-DLL applications, really can be quite massive, and we're talking about a level of complexity that many Windows applications just never approach because of the size of these applications, the number of procedures. If you were to try to duplicate all of this in say C++ code, I mean, you know, just imagine a 500 procedure, you know, eight DLL, which is not that big an application, imagine something like that written in C++ and what a nightmare that would be to manage. So, as you say, there are an awful lot of things that have to happen for a clean shutdown and it sure is nice to see some templates to handle that stuff.

Andrew Guidroz: And personally, I prefer the old method. Most of the applications that I'm doing, I want the client to finish doing what they're doing before they actually shut the computer down, so the fact that by default, my Clarion app doesn't go down, it makes the user go through his normal process of finishing, that's better for me in 99.9% of what I do. But there are occasions that I do want, when I've got a little dummy apps that run and don't do a whole lot, you know, but most of the apps that I handle are data entry type apps, so in those cases, I really want the user to finish his train of thought.

Dave Harms: Yeah.

Andrew Guidroz: But conversely, little things like I've done for television stations, I'm gonna shut down the computer, who cares if the video on air dies because if they're shutting the computer down, I mean they're wanting everything to go down.

Dave Harms: I see, yeah.

Andrew Guidroz: <laughs>

Dave Harms: And really, it's a template, it's not something, it's not a change in behavior,

it's a new option, so you will enable this for the apps you want to shut down automatically and you will leave things the way they are for the apps that you don't want to shut down automatically.

Andrew Guidroz: You got it.

Dave Harms: Yeah.

Andrew Guidroz: Now they even have some new support for the people who are, uh, who are stuck in the past, no that's not a nice way to put it.

Dave Harms: <laughs>

Andrew Guidroz: Some people like the.

Dave Harms: The people who are forced to use Legacy by circumstances beyond their control?

Andrew Guidroz: Yeah, that's a nice way to put it. Let, you know, there is now trigger support. The dictionary triggers that you can define. Some people were complaining about the fact they can't use them in their Clarion templates. Well now it works.

Dave Harms: Yep.

Andrew Guidroz: I mean, they've handled it, they've added a new trigger support property on the file control tab, in the IDE, you click that, you go to town, the thing works. That's pretty slick. Another Legacy thing they've added is the Enter by Tab template. They've enhanced it a whole bunch where, you know, Windows doesn't like to use Enter, it likes to use Tab, Tab here and Tab there, but they've added even more things now to clean up that whole interface, that if you want to use the Enter key like QuickBooks does, it's what all my clients like to see. It's "Man, can't you make it work like QuickBooks?"

Dave Harms: Mm-hmm, yeah.

Andrew Guidroz: This gives you one more way to do that, to get that old DOS styled interface going. So that's pretty cool too.

Dave Harms: Yeah. And then we've got some changes to the runtime library, we've got some changes to the compiler, linker and some changes to the IDE. I mean, these are not

really major things. There is substantial change to the debugger. I mean, I guess the debugger more or less looks the same but now there's...

Andrew Guidroz: It's a little more thread friendly for all these various threads that you have going on.

Dave Harms: Yeah, there's now a thread list you can bring up.

Andrew Guidroz: Right, right.

Dave Harms: Yeah, so and then there's also a movie you can watch to get a better idea of how that actually works, and this is gonna make Russ Eggen very happy.

Andrew Guidroz: Mr. Debugger, that's right.

Dave Harms: Mr. Debugger.

Andrew Guidroz: I don't use the debugger much but some people do. I've had to use it from time to time and usually when I'm in there, man it's a painful experience, not just from using it but from what I've got wrong.

Dave Harms: Yeah, I can't say that I'm really a big debugger fan. In fact, I'm just gonna use that as a little segue here and mention last week in Clarion Magazine, I published a sort of a little [mini review of RADProfiler](#), which for those of us who are a bit debugger-phobic, is I think a really awesome way to do some debugging. It actually builds on some code that I wrote way back when using the profiler hooks, and but it also gives you the option of outputting stuff to DebugView, which is a really cool tool from a company called [Sysinternals](#) and so little plug for the article and for RADProfiler. It's a really slick piece of work and it's free.

Andrew Guidroz: Well here's another new feature that I like and I used to have to fight with and it was kind of painful is that they've expanded a little bit of the file manager to have some new methods to let you change the open mode on a file. What happened to me was you'd write the app and you need to open the file or share. Well sometimes you didn't want to open the file shared, you wanted to open it exclusively to do some nifty stuff and then turn around and open it shared. It wasn't so easy to do under ABC. It was really a little bit of a pain to switch file modes. Well, they've added this...file opening modes. They've added these two new methods called set open mode and get open mode so that you can say "Al right, change this thing to be exclusive." That way I can go and do some

heavy duty updating and keep everybody locked out while I'm doing it and then turn around and open it back shared. So that's very slick for me when I'm doing some real intensive file I/O. I mean for instance, in some of the accounting apps that I've written, I like to let people get in and post things into the accounting with TPS files but when it gets down to closing the books for the month where I'm doing some updating and of historical figures and those type of things, I wanna open exclusively and basically make sure no one's in there posting accounting entries while I'm rolling those history figures. So that's pretty slick.

Dave Harms: Right.

Andrew Guidroz: Much easier.

Dave Harms: Another thing that maybe we should mention is the BreakManager class has some enhancement. You've got two new total types, Sum Prod and Weighted Aerge, so sum of the product and...

Andrew Guidroz: Some of the product, yeah, yeah.

Dave Harms: Some of the product, weighted average, there you go.

Andrew Guidroz: <You got it brother.

Dave Harms: Yeah. Lots of.

Andrew Guidroz: Now, there are also a bunch of bug fixes, you know, the first stuff we've talked about here, that's all enhancements and additions to the product.

Dave Harms: Mm, yeah.

Andrew Guidroz: So that's where you've gotta sit down and look at it and what a lot of people are going to say is "I don't want to pay for bug fixes".

Dave Harms: Yeah, this is what's gonna happen.

Andrew Guidroz: You gotta give me some sizzle, you know, if you're gonna give me a product and you're gonna make me pay for it, give me something besides just my bug fixes. You've got a lot of functionality out there. You've gotta decide if you think it's cost effective for what you're doing, whether or not you need the new features or not.

Personally for the SQL side, just that for me is worth it. I've gotta get in. Some people the Relation Tree thing I think will be cool for them.

Dave Harms: Mm-hmm, yeah, for me it's gonna be the SQL stuff too.

Andrew Guidroz: Yeah, yeah, and I mean, maybe you're a debugger kinda guy and you like that, but there's also a bunch of bug fixes, I mean, probably 35 or 40.

Dave Harms: And this is...

Andrew Guidroz: Off the top of my head in just the templates and classes, I mean, then you go into just, you know, they've got improvements, they've got runtime functionality, they've added about two or three dozen things, they've added runtime window handling has had a couple of dozen things changed and updated and the compiler and linker has got what looks like 20 more updates. And I mean there's a lot out there.

Dave Harms: Yeah, yeah, it's a pretty big list. But that of course raises this whole specter of inevitably there are gonna people who are gonna say "well, you know, I've paid for Clarion 6, I've been getting patches, these are bug fixes, why can't I get these bug, why am I having to pay for bug fixes?"

Andrew Guidroz: Right and some are gonna say I'm spending 99 bucks, how does that guarantee that I don't have more bugs, and it doesn't, it really doesn't. So that's when you started looking at another part of the new product, the subscription programs that they're releasing and maintenance programs.

Dave Harms: Yes.

Andrew Guidroz: Now, okay, so we talked about, let's see, let's back up. You've got 69 bucks was the price for the Relation Tree Template. That's the cheapest thing that they're selling right now, 69 bucks. Then and all these prices of course are U.S. dollars in case you're all wondering out there for you Canadians that are, that's like a million dollars in Canadian money, 69 bucks.

Dave Harms: Well that's how it used to be, now it's kinda like about 1575, \$15.75.

Andrew Guidroz: <laughs>

Dave Harms: Y'all need to do something about your currency I think, it's hurting.

Andrew Guidroz: <laughs> It's not going well but, so you've got 69 bucks for the Child Template. If you go the Relation Child Template. If you go 99 bucks, you get that, you get all these other new features and all these bug fixes, but it's a one shot there as well. You spend 99 bucks and you get it and that's it. If you want to go a little...

Dave Harms: Although presumably there would be patches to 6.2, as there have been patches to 6.1, so.

Andrew Guidroz: Presumably.

Dave Harms: Presumably.

Andrew Guidroz: From there you go to what they call the simple maintenance program. The simple maintenance program is 150 bucks. Okay, so you're looking at another 51 dollars over the price of 6.2.

Dave Harms: Mm-hmm.

Andrew Guidroz: In lieu of, they're not cumulative, you know, add one and then the next and the next, you can either spend 69 bucks for the child stuff or you can spend 99 bucks for your upgrade to 6.2 or you can go all the way to 150 bucks, and what 150 bucks gives you is a, you'll get all the dot release upgrades and all the maintenance releases, you'll get new release notifications via e-mail, you will get educational discounts, you will get discounts on any new products that SoftVelocity has coming down the pipe, and there are rumors of quite a few that they've discussed at DevCon and we've discussed a few through the .Net product here on the pod cast and the C7 product, and what else. Oh, you get access to a new thing, the other new thing they've come up with, the Problem Tracker Developer Support System. Now this is a brand new thing. You don't have to spend any money to get into this thing.

Dave Harms: Now this is what, somebody was saying I think somewhere, maybe third time's the charm on the bug tracking system?

Andrew Guidroz: Yeah, they tried it years ago and it didn't work. Then they tried like a user group, news group type thingy with a weird interface and that didn't work very well either, so they've reinvented the wheel again <laughs> and come up with a totally new way to go about it, and this new system, you can submit your incident reports, you can look at all open and confirmed bugs, anyone can, you don't have to be a, you know, you can be a 5.5 user and haven't spent any money for any upgrades. You have access to this

thing. But being a subscriber to that simple maintenance program for that 150 bucks, you'll get tracking numbers and you can do things like fine out the progress of your bug by actually talking to somebody with SoftVelocity. The extra, what did we say 51 bucks, the extra 51 bucks basically buys you access to putting a bug in somebody's ear in SoftVelocity's offices. You're gonna get more access to interact with them. Currently, there's not a lot of interaction with bugs. You basically, you e-mail them, it's sort of a black hole, you look at the next fix list and you hope every thing's okay.

Dave Harms: It's gonna be interesting to see how this works out. I think what I foresee happening in the short term anyway, I don't think it'll be a long-term issue, but in the short term, I would expect the standard hue and cry about "we're owed these bug fixes, you can't charge us for this" yada, yada, yada, and I'll grant that there's a certain, I mean there is a certain logic to that, and it is always irritating when there's a bug that's biting you particularly badly and it hasn't been fixed and then somewhere along the line you discover that oh, by the way, you're gonna have to pay in order to get the fix to that bug.

Andrew Guidroz: Right.

Dave Harms: And, you know, I don't really want to on the one hand minimize that, because I think it is gonna be a bit of an issue for some people. I think it's probably going to be the biggest issue for people who are stressing the product the most, because certainly in my use and I think I don't really, most of the time I don't stress it very hard. I found 6.1 to be quite stable and I haven't run into any significant problems with it. Somebody who is really pushing things to the max is obviously going to be bumping into more issues and some of those people might be somewhat unhappy about having to pay for bug fixes.

Andrew Guidroz: But certainly if you're stressing the product that much, the fact that you can pay in now to get more interaction with SoftVelocity's offices, that's worth some money, I mean, to me when I've really pushed the core of the product, and I've done it over the years that I've used it, from time to time they've seen my frustration and SoftVelocity has reached out, you know, and said, you know, "look, here's", you know, they'll pop an e-mail to me that says "Look, we see you struggling with this thing, we see it's a bug, we're having problems recreating, give us some information." They don't guarantee that they do that and they don't do it a lot, because I mean, it's expensive. I mean, there's no question <laughs> that, you know, sending a developer to start dealing your clients head to head can be a very expensive proposition, and it takes away from some of the new development, so...

Dave Harms: Sure, and that one developer is worth a lot per hour when you think about if they're a good developer.

Andrew Guidroz: Right, so they've got to figure out a good balance there, but with this, you got a little more pull with them, a little more interaction with them. So you're not paying to submit the bug, you're not paying for the bug to be fixed really, you're paying for the access, it's access to SoftVelocity.

Dave Harms: Yeah, and certainly it's a good way to identify in a way, you know the people who are serious about getting, about helping you get stuff fixed are certainly going to be wanting to sign up and pay a relatively small amount of money to be part of this system, so it may help to improve the signal to noise ratio a little bit as well.

Andrew Guidroz: Right and the 150 bucks, if it's gonna cost you 100 bucks, 99 dollars to upgrade to 6.2 anyway, the 50 bucks is basically prepaying you, you know, some companies 50 bucks would only buy you a single support incident. I'm not seeing that in this and from what I can understand, if you run into problems, you post them, you'll get tracking numbers and status updates as to what's going on. I do understand that the free version of this thing, you know, when you submit your bug, you'll get notification that your bug's been accepted and you'll also get notification when that bug has been fixed.

Dave Harms: So it's important to note that you do not actually have to be buying the maintenance program to submit bugs.

Andrew Guidroz: You don't have to be a subscriber, right.

Dave Harms: Anybody can do that.

Andrew Guidroz: Exactly, exactly, but you.

Dave Harms: But you won't get your hand held.

Andrew Guidroz: Right, you won't get your hand held, you won't get told the status as things are progressing, and this is the simple, what they call the simple maintenance program. Now, what about..

Dave Harms: Well, before we do that, I just wanna say.

Andrew Guidroz: Yeah?

Dave Harms: I just wanna say one more thing about the whole issue of bug fixes.

Andrew Guidroz: Yeah, yeah.

Dave Harms: I mean, you know, I will say I do expect that there'll be some screaming and kicking about paying for 6.2. I think that if you look at, I mean, I don't know that I've ever bought an upgrade that didn't have bug fixes <laughs> from the previous version, you know. I mean there are always, I don't think there's ever a release of any substantially complex program that is completely bug free in between releases. So it's inevitable. I mean, with Microsoft Office or with, you know, with any, like even with those mainstream kinds of applications, you know, there will be people who will upgrade to the new version because finally this thing is fixed or finally that thing works the way it's supposed to, so I don't think this is an unusual situation.

Andrew Guidroz: No. From the simple maintenance program they've got the new super duper product which they call the core...

Dave Harms: The super duper maintenance program.

Andrew Guidroz: That's right. The core subscription program.

Dave Harms: Yeah.

Andrew Guidroz: Now here's where we start talking about spending some more money and the pricing at least initially as an introductory type thing, for Professional addition users, they're gonna price this thing at 350 and if you're in Enterprise it's gonna be 750.

Dave Harms: And after June 15th, those go up to 400 and 800 respectively.

Andrew Guidroz: Right. Now what does it get you? Now here's where I'm a little more excited, okay.

Dave Harms: <laughs>

Andrew Guidroz: You're gonna get not just maintenance releases and the dot releases, okay, you're gonna get all major upgrade releases, so...

Dave Harms: For a period of 12 months.

Andrew Guidroz: One year.

Dave Harms: One year.

Andrew Guidroz: One year, and you're guaranteed one major upgrade for your year's subscription, okay.

Dave Harms: Which means...

Andrew Guidroz: Which means that within 12 months of when you cut that check, they are attempting to make sure that the next major revision is coming and we know what the major revision numbers mean.

Dave Harms: <laughs>

Andrew Guidroz: It means we're gonna go from 6. something or other to 7. Right?

Dave Harms: Right.

Andrew Guidroz: Right, that's major. That's a major release.

Dave Harms: That's major, yeah. This is absolutely. And this is one we haven't seen, we basically haven't seen a peep out of C7 yet. Well I guess we saw a couple of screen shots back in DevCon.

Andrew Guidroz: At DevCon.

Dave Harms: But nobody within the development community has been playing with any code yet, and we've all been waiting somewhat anxiously, and I guess there are some people out there who are saying, you know, wondering where it is and so this is, as you say, a very strong indicator that they are at least planning to have 7.0 released within 12 months.

Andrew Guidroz: That's right. Now if something should happen of course and things take longer to get done.

Dave Harms: You will still get your 7.0 release.

Andrew Guidroz: The wording is you're still gonna get it, that's right. So, and you're also gonna participate in the beta.

Dave Harms: Mm-hmm. The C7 beta?

Andrew Guidroz: That's what the subscription will. The C7 beta.

Dave Harms: There's actually an absence of information here about Clarion.NET, so it would appear that at this point Clarion.NET would be a separate...

Andrew Guidroz: Totally separate product is what it looks like currently.

Dave Harms: Yeah, so the core subscription program does not in fact cover Clarion.NET, which is, and Clarion.NET is really positioned as a separate product in any case, not as an upgrade.

Andrew Guidroz: Right.

Dave Harms: It's not meant as an upgrade for C7, although they will share the same IDE as far as we know.

Andrew Guidroz: As far as we know.

Dave Harms: Mm-hmm, yeah. And I would say certainly that the core subscription program is another indicator if they plan to have something released within a year's time, then you know that a beta has got to be getting pretty close, because...

Andrew Guidroz: Right.

Dave Harms: ...you're really look at, for anything of the kind of complexity that we're talking about in a full Clarion release, you've gotta be thinking somewhere in the neighborhood of a year for a beta program anyway.

Andrew Guidroz: Right, so you're looking at, you know, if you're an Enterprise guy and you're spending all the duckies, you're looking at 750 dollars. That's gonna get you 6.2, that's gonna get you into this maintenance program where you can see the bugs and you can get people to hold your hand over at SoftVelocity when you have a bug, they'll tell you, you know, how long, how far along your bug is coming and give you status reports and.

Dave Harms: How big your bug is growing.

Andrew Guidroz: How big your bug is growing, yeah. <laughs>

Dave Harms: Whether it's been laying any eggs or not and.

Andrew Guidroz: That right, and...

Dave Harms: And whether they're having to bring in the fumigators or they're just fleeing the offices.

Andrew Guidroz: Then if 6.3 should come out in the next two months or three months, you're still locked in, you still can get access to that, and if 7.0 goes into beta, you're locked into that, and if 7.0 goes gold, you're locked into that as well. So, I mean, there's some positives coming down the pipe here, you know. And when you start looking at it, I don't remember what the beta's cost of Enterprise way back. It's been so long since the full blown betas and all that, but you had to pay some substantial money to participate. It was basically the price of an upgrade and then you participated in the beta. Well with this, you're not far off. You're still in that same ballpark. You're still doing some of the same things, so I'm not alarmed.

Dave Harms: Yeah, I would, you know, I mean I guess the first thing, you know, a lot of people might be inclined to do when they look at these numbers is go and compare to other similar development tools and if you go and look at some other development tools, I don't think you're seeing anything that's really out of line here.

Andrew Guidroz: No, I don't think. I think, you know, you're, 'cause I went looking at competitive products that shall remain nameless. <laughs>

Dave Harms: <laughs>

Andrew Guidroz: And yeah, I mean, the pricing schedule seems to match up pretty well. I mean, it's very, very similar. About the only guys out there that don't seem to match exactly is Microsoft, but then again, if you.

Dave Harms: They're in a world of their own. <laughs>

Andrew Guidroz: <laughs> If you own Office and you own Windows, well you can do things a little bit differently.

Dave Harms: Yeah.

Andrew Guidroz: You can make the X-Box and not make a single nickel on any devices, you can just, you know, give them away and try to make money off the games. You know, I mean they just have a different model that no one else has. But no, I think this is a positive thing. I think that the fact that they seem to be positioning themselves to make a run at 7 is, you know, that, <laughs> someone asked me today as a matter of fact about 6.2, what did I think the most exciting thing about 6.2 is, and I think the most exciting thing about 6.2 is that it means that they're starting to focus on 7.

Dave Harms: Yeah, yeah. In fact, I was a little bit, when I first, you know, there was sort of this buzz about a coming release and I was hopeful that we'd may be seeing an, you know, an alpha or an early beta or something like that of C7, and I confess to being a little disappointed when it turned out that it was gonna be 6.2 and but as you say, this really does open the door for 7 and they've given some pretty clear indicators that, you know, the next item on the list is 7 and I'm sure there will be some, there will probably be bug fixes and patches and so forth to 6.2, as there is with any release. Your 6.2 developers at this point, I would say most of them should be freed up now to work on C7 and maybe on Clarion.NET.

Andrew Guidroz: Yes, sir.

Dave Harms: And...

Andrew Guidroz: And you're excited about the Clarion.NET part of it.

Dave Harms: I am, I tell you, you know, the more I read about .NET...

Andrew Guidroz: I'm not as excited about .NET, although I do think .net is cool. From my standpoint, I'm more excited about the fact that they're opening up the IDE to support it, which means that they're gonna have to support COM a lot better than the IDE and what that means to my pure Windows desktop type applications is that they're gonna be able to talk and shake hands with stuff in the .NET world a whole lot easier. I'm not so concerned about writing a true .NET app as I just want my Clarion apps to play nice with them.

Dave Harms: Mm-hmm. I'm really interested to see what the new C7 when we actually get out hands on it, you know, and to find out how much of this is Clarion code and how much is C code and the thing is, is that Bob has indicated in one little posting he made in the C7 newsgroup that there was some .NET stuff in the Clarion 7 IDE, like some of the IDE itself appears to be written in .NET.

Andrew Guidroz: Written in .NET. I got that impression, yeah.

Dave Harms: And I find that really fascinating, man, it sort of makes me wonder how much of the Clarion 7 IDE might be written in .NET. Certainly if a lot of it's written in Clarion and you have a Clarion.NET compiler, you know, I think that certainly down the road at some point I would expect the Clarion IDE to be a .NET application and, you know, that's...

Andrew Guidroz: And what does that imply as far as cross platform, you know, I mean there are all kinda things that [unintelligible] and start to popup.

Dave Harms: Yeah, yeah, there're just all kinds of things that start to come up then. Well, and you talked about liking the C7 IDE because it's gotta have better COM interoperability and to that I would just say I think that's great, I think we need better COM capabilities in Clarion, but I've gotta say, the thing that makes me long for .NET the most is COM <laughs> because I still find COM a complete nightmare most of the time.

Andrew Guidroz: It ain't easy.

Dave Harms: It ain't easy. And when you've got something you wanna do, boy, you know, it can be a real challenge to try to find out exactly how, what a COM object can do and how you're gonna use it and, you know, they really didn't make it easy with that one. I mean, .NET has substantial improvements and refinements over the whole business of component reuse, and we have learned. I mean, language developers, designers have learned a lot in the years between COM and JAVA and they've also learned a few things between JAVA and .NET and I think Clarion developers certainly stand to benefit from a lot of that. I mean, the current survey I've got going right now, which is how do you describe your knowledge of COM in general, and over half say nonexistent and 27% beginning and intermediate 11%, 1.6% say expert and so you've got about 12% of Clarion developers who answered that poll who feel like they have some, you know, modest to good ability to use.

Andrew Guidroz: Well, those of us who read your magazine, Dave.

Dave Harms: Yes, exactly.

Andrew Guidroz: We've done at least a little bit with it.

Dave Harms: Yeah, but even there it's only, even with the benefit of Clarion Magazine,

you know, it's a pretty small percentage, and I think...

Andrew Guidroz: Read, learn, read, learn and solve.

Dave Harms: Read, learn and solve.

Andrew Guidroz: And you can still, you can still get the [Clarion Magazine coffee mug](#) over at your web site.

Dave Harms: Exactly right, yeah.

Andrew Guidroz: With the phrase Clarion Magazine - Read, Learn and Solve.

Dave Harms: And on the obverse, would that be the obverse side of the mug?

Andrew Guidroz: Right, you got the obtuse information...

Dave Harms: You get the obtuse information on the obverse side. <laughs>

Andrew Guidroz: <laughs>

Dave Harms: That's where you'll find your Planet Clarion logo.

Andrew Guidroz: There you go, that's right, we get to plug some product here, man.

Dave Harms: Yeah, it comes complete with a list of key words to memorize for the next podcast.

Andrew Guidroz: And hopefully the next time we'll have our Planet Clarion espresso maker...

Dave Harms: <laughs>

Andrew Guidroz: ...ready to be sold. It'll be a little bit more than the Enterprise subscription rate. <laughs>

Dave Harms: Yeah, that's right, that's right. <laughs> But we'll ship it anywhere in the world for free.

Andrew Guidroz: Anywhere. South Africa.

Dave Harms: <laughs> South Africa, that's a very nice [picture](#) of...

Andrew Guidroz: Of your Planet Clarion and Clarion Magazine mug by Gary James.

Dave Harms: Yeah, by Gary James, yeah.

Andrew Guidroz: I haven't seen Gary James, it must be six or seven years is the last time I've seen him.

Dave Harms: Oh really, well you should've been at the last DevCon, because he was there.

Andrew Guidroz: The last time I saw anybody from South Africa it was with Jono. Jono was at the last ETC.

Dave Harms: Yeah, that's not that long ago.

Andrew Guidroz: Oh, not long ago at all.

Dave Harms: And the famous towel incident.

Andrew Guidroz: Hey.

Dave Harms: Yeah. Good thing this is radio.

Andrew Guidroz: We have pictures.

Dave Harms: <laughs>

Andrew Guidroz: Y'all wanna look at this picture.

Dave Harms: Oh, my goodness.

Andrew Guidroz: It's terrible. <laughs>

Dave Harms: Uh-uh, that's no good.

Andrew Guidroz: Anyway.

Dave Harms: We can't show that.

Andrew Guidroz: So what else we got Dave? 6.2 I think that's a lot.

Dave Harms: That's a lot, and if you have particular things that you'd like to hear us talk about or particular people you'd like to hear us offend, just drop us an e-mail, planet@clarionmag.com and we'll see what we can do, and hopefully we'll have Bob Zaubere on again soon.

Andrew Guidroz: Yup.

Dave Harms: He's gonna be kinda busy for the next little while with a trip down to South America and, but when he gets back and has a chance to get caught up, hopefully we'll be able to talk to him about some of the new stuff and maybe by then we'll be able to talk about a beta of some new product. Who knows.

Andrew Guidroz: You got it.

Dave Harms: If we get lucky. Okay, we've got a little something different. Usually we close out our show with the same track that we start the show with, but we're gonna do something a little bit different. We've got...

Andrew Guidroz: A lot different.

Dave Harms: A lot different. We've got a couple of tracks here, well, we've got one track here we're gonna play some classical guitar by Pierre Tremblay. Pierre is one of the developers working for SoftVelocity.

Andrew Guidroz: French Canadian.

Dave Harms: French Canadian, and...

Andrew Guidroz: Comment ca va?

Dave Harms: The man has a serious gift with the guitar and so I think we'll just leave it at that and we'll play out. You can listen to some of Pierre's...

Andrew Guidroz: Right, even if you don't speak French.

Dave Harms: Even if you don't speak French, you might...

Andrew Guidroz: When he plays the guitar, it still sounds a lot like the English version.

Dave Harms: That's right, that's right.

Andrew Guidroz: <laughs>

Dave Harms: He's doing dynamic real time translation with this I think. Yeah, all right. Here's Pierre.

<music>

Reader Comments

[Add a comment](#)