

Clarion Magazine

Clarion News

- » [C7 IDE Movie Shown At Brazilian DevCon](#)
- » [Dang! Hillbilly On First!](#)
- » [Clarion 6.2 Now Available](#)
- » [EasyCOM2INC TXCTRL 1.00](#)
- » [Window Settings 3.2](#)
- » [xDataBackup Manager Lite v2.1](#)
- » [Dennis Baggott's Clarion Toolbox](#)
- » [Taboga Report's Data Exporter Library Beta 1](#)
- » [Web Email Cloaking Utility](#)
- » [BoTran 2.1](#)
- » [EasyCOM2INC_SQLDMO 1.00](#)
- » [Blog: Gary James](#)
- » [dpQuery 2.04](#)
- » [ARD Report Writer Compatible With C6.1](#)
- » [Simpad Templates Updated](#)
- » [EasyCOM2INC 2.04](#)
- » [ezFormFiller 1.0.6](#)
- » [Pre-Purchase Clarion 6 Tips & Techniques Print Book, Save 40%](#)
- » [Clarion 6.1 HotFix 9034](#)
- » [SB5 Developer/Professional Edition \(RC3\) Build #1130](#)
- » [ezFormFiller Beta Handles PDF Forms](#)
- » [CWPlus 3.02](#)
- » [Resume Upload/Download Area](#)
- » [SkyHi Debug Templates 2.0](#)
- » [EasyCOM2INC 2.03](#)

[\[More news\]](#)

Podcast



Clarion 6.2 has been released, and Andrew and Dave talk about the new features, the subscription program, the upgrade fee, and Clarion 7. 42:17:00, 15224K

 [Listen now](#)

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

Latest Free Content

[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.

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

Clarion 6 Tips & Techniques book Coming soon. Pre-purchase now and save!	 CLARION MAGAZINE
--	--

Latest Subscriber Content

[Planet Clarion Transcript: Bruce Johnson on the IP Driver](#)

Our interview with CapeSoft's Bruce Johnson about the IP driver remains one of our most popular podcasts. If you've ever had to deal with bandwidth issues or file corruption using TPS files on a network, this transcript is a must read.

Posted Wednesday, May 04, 2005

[PDF for April 2005](#)

All Clarion Magazine articles for April 2005 in PDF format.

Posted Wednesday, May 04, 2005

[Modifying the Common Dialogs: How Hard Could It Be?](#)

Have you ever wanted to add a button or other control to one of the built-in common dialogs for opening a file, saving a file, or printing? Jim Kane thought it would be nice to add a multi-page TIFF preview button to the file open dialog. How hard could it be?

Posted Thursday, May 12, 2005

[A Better DosFileLookup](#)

When some standard Clarion functionality doesn't meet your need, what do you do? First, you embed some code. Then, when you need that code elsewhere, you start thinking about how to avoid duplicating the embeds. As Dave Harms shows, often the easiest solution is to create a custom, derived ABC class, and tell the templates to use your class instead of the ABC original.

Posted Friday, May 13, 2005

[Whitemarsh's Use Of Mimer With Clarion](#)

If anybody in the Clarion community knows about SQL standards, it's Michael Gorman. Read why Mike's company, Whitemarsh Information Systems Corp, chose the Mimer database for its own Clarion-built metabase product.

Posted Thursday, May 19, 2005

[Easier Version Control: Using TortoiseSVN With Subversion](#)

The two heavyweights in the open source version control world are the old workhorse, CVS, and the new contender, Subversion. Mark Geisinger shows how easy it can be to get started with Subversion, using the Explorer-based TortoiseSVN client.

Posted Thursday, May 19, 2005

[First Look: RADventure's RADProfiler Simplifies Tracing and Debugging](#)

Clarion has long had the ability to generate a trace of all procedure calls in any given application. Radventure's RADProfiler adds some neat twists to previous efforts to exploit this technology, including trace filtering and the option to send output to SysInternals' DebugView utility.

Posted Friday, May 20, 2005

[Generic DLLs The Template Way](#)

A truly generic DLL can be difficult to achieve using the standard templates. Hand-coding the DLL, as Jeff Slarve recently described, is one option; another is to use a specialized template to manage the creation of the export file. Bjarne Havnen shows how it's done.

Posted Thursday, May 26, 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 Tuesday, May 31, 2005

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

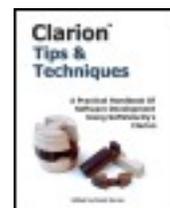
Printed Books & E-Books

[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 two 600+ page softcover books: [Clarion Tips & Techniques](#), and [Clarion Databases & SQL](#). These books are packed with information useful to any Clarion developer. 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](#)

[New C6.2 Help Files](#)

An install program containing the new 6.2 help files is now available on the SoftVelocity home page. There are also some new example programs showing how to use the new Multi-Child Tree Template.

Posted Friday, June 03, 2005

[NiceTouch Products 6.2 Compatibility Testing](#)

The following NiceTouch products are being compiled and tested within the next several days for 6.2 compatibility: Query Wizard; Report Wizard; View Wizard; SpreadSheet Wizard; CrossTab Wizard. All TimeSavers products should be compatible with no changes, as they do not use any DLLs. ReportDAT will need to be tested and recompiled.

Posted Thursday, June 02, 2005

[xFastButton Template 1.0](#)

SealSoft has released xFastButton Template 1.0. This global extension template lets you globally set properties for all buttons in your application. You can set main buttons properties: Text, Tips, Icon, Size (Width and Height), Justification, Flat, Skip, Scroll. Supports single EXE, Multi-DLL (local mode, standalone mode), 32-bit. For Clarion 6.1 and Clarion 5.5. Price: \$49.

Posted Thursday, June 02, 2005

[xQuickFilter 2.15](#)

xQuickFilter 2.15 is now available. Changes include the ability to choose how an item with a name of a field in xQuickFilter popup-menu will be formed: Field's Description (from dictionary), or; Field's Column Heading (from dictionary), or; Listbox header (i.e. ?Browse:1{PROPLIST:Header, nn}). Compatible with Clarion 6.1 (build 9033).

Posted Thursday, June 02, 2005

[Free .NET E-Books](#)

Carl Barnes recommends two free e-books from APress: Programming VB .NET: A Guide For Experienced Programmers, and Dissecting a C# Application: Inside SharpDevelop.

Posted Thursday, June 02, 2005

[Das Products C6.2 Compatible](#)

Clarion 6.2 compatible versions of all Das products are now available.

Posted Thursday, June 02, 2005

[xToolTip 1.8](#)

xToolTip 1.8 is now available. New in this version: Processing of Critical Section for more stable work under Clarion 6.1; Compatible with Clarion 6.1 (build 9033). New installation kit for all Clarion version (except 6.2).

Posted Thursday, June 02, 2005

[New vuDeploy Help File](#)

vuDeploy has had some major enhancements recently, as documented in this new help file. You can download a fully functional 30 day trial version. Although there is a splash screen when you start vuDeploy, the EXEs you produce do not contain any splash screens or notices (unless you put them there), and will never expire. vuDeploy creates a self-extracting Windows 32 bit EXE file that acts as a container for any number and type of files you wish to include in it. When the EXE file is run, all files are automatically extracted and written to disk. When creating this EXE file, you can specify a number of actions you would like to have performed. This includes: Automatically running a selected file that was extracted (either an EXE or any file that has an association); Automatically compress and decompress files within the EXE (save disk space); Display a user defined message after file extraction; Password protect the EXE; Allows file modification; Set the "Hidden" attribute of all extracted files; Set the "System" attribute of all extracted files; Automatically delete all extracted files when the auto-run file or displayed message

terminates.

Posted Thursday, June 02, 2005

[Online Theft Recovery Service](#)

Scott Ripley has launched a new service developed with C6 and NetTalk.

TheftRecoveryOnline helps track computers attached to the Internet, so if your computer is ever stolen you have a better chance of getting it back.

Posted Thursday, June 02, 2005

[EIP Queue Demo](#)

A demo of using EIP with a queue is now available at the Par2 download center. The demo contains examples of edit controls, a spin box, a drop list and a calendar lookup, how to replace controls at run time based on a condition and some basic setup for the EIP.

Posted Thursday, June 02, 2005

[PDF-XChange 6.2 Compatible](#)

PDF-XChange PRO and Tools SDK's (build 3.4081) have been tested and found to work with Clarion 6.2.

Posted Thursday, June 02, 2005

[iAlchemy Skin/CYA 6.2 Compatible](#)

There are no know issues with iAlchemy Skin and CYA after initial compile and tests with 6.2.

Posted Thursday, June 02, 2005

[xPathManager 1.3](#)

xPathManager 1.3 is now available. This release fixes a but with file creation.

Posted Thursday, June 02, 2005

[PowerOffice Products 6.2 Compatible](#)

PowerOffice products are all source code so there should be no problems with using them with Clarion 6.2.

Posted Thursday, June 02, 2005

Riebens Systems Products 6.2 Compatible

It seems all Riebens Sytems Products run under C6.2.

Posted Thursday, June 02, 2005

Comsoft7 Products Expected To Be 6.2 Compatible

The entire line of Comsoft7 products is all source code, so they should have no problems running with C6.2

Posted Thursday, June 02, 2005

IceTips and Clarion 6.2

All IceTips products should compile in 6.2. The installs however do not support 6.2 yet but builds will be added in the next two weeks or so. If you run into problems with any of the IceTips products in 6.2, then please email Arnor or leave a message on the Icetips Bulletin board.

Posted Thursday, June 02, 2005

iQ-XML For Clarion 6.2

A patch for iQ-XML for Clarion 6.2 is now available. Be sure to read the text file inside the zip and make sure you download both the 6.1 and 6.2 patch and install in that order.

Posted Thursday, June 02, 2005

Epsilon Hosting Offers New Plans

Epsilon Hosting now features new plans, more support, and an all new look. The price is still \$7.77 a month with 24/7 support. To celebrate these changes, Epsilon has a special offer for Clarion developers: purchase any annual hosting account, and get a domain name registration for free. Purchase any plan above the starter plan and get the basic marketing plan at no extra charge. E-mail robbly at epsilonconcepts.com for more information.

Posted Thursday, June 02, 2005

RADRestyle Trial Version

A trial version of RADRestyle is now available. RADRestyle is a template with collection of classes wrapped into a DLL that can restyle your application at runtime. RADRestyle is €49.00.

Posted Thursday, June 02, 2005

[Blog: Michael Lawson](#)

Clarion developer Michael Lawson's blog covers a lot of subjects, including, occasionally, Clarion.

Posted Thursday, June 02, 2005

[C7 IDE Movie Shown At Brazilian DevCon](#)

Attendees at the VI Condev Mercosul Gescla conference were shown a movie of the new Clarion IDE on Friday. One attendee described the new IDE as "very slick, very MS Studio in lots of ways." The demonstration did not include the templates/AppGen, which is still under development, but did have features such as a workable form designer. The IDE compiles Clarion.NET, C#, and VB.NET. No specific release dates were mentioned.

Posted Saturday, May 28, 2005

[Dang! Hillbilly On First!](#)

RPM and AFE for Clarion 6.2 are now available from Lodestar Software.

Posted Friday, May 27, 2005

[Clarion 6.2 Now Available](#)

Clarion 6.2 has been released, and is a free upgrade for anyone who bought or upgraded a license for PE or EE after November 1, 2004. Several options are available for other developers, including flat \$99 fee and two levels of subscription program. New features include enhanced SQL functionality, new core templates (multi-child relation tree, shutdown with Windows, clean closedown of multi-DLL apps, minimize to system tray, trigger support in the Clarion chain), thread enhancement for the debugger, EIP, RTL and RTF improvements, and a variety of other enhancements and bug fixes. Look for a discussion of 6.2 in the next Planet Clarion podcast, available Wednesday, May 25.

Posted Tuesday, May 24, 2005

[EasyCOM2INC TXCTRL 1.00](#)

Based on COM and EasyCOM2INC technologies, this product demonstrates the use of the TX Text Control 11.0 ActiveX (<http://www.textcontrol.com>) in a Clarion application. Requires Clarion 6.1 9033 or higher; EasyCOM2INC 2.04 or higher; TX Text Control

11.0. Price: \$60. This is free for all registered customers of EasyCOM2INC.

Posted Tuesday, May 24, 2005

Window Settings 3.2

New in Window Settings 3.2: Errors with program hanging and error with opening xWS ini file under Clarion 6.1 (New THREAD model) were fixed; Compatible with Clarion 6.1 (Build 9033).

Posted Tuesday, May 24, 2005

xDataBackup Manager Lite v2.1

New in xDataBackup Manager Lite v2.1: C6 version compiled under Clarion 6.1 (Build 9033); Error with creating file list for backup was fixed.

Posted Tuesday, May 24, 2005

Dennis Baggott's Clarion Toolbox

Dennis Baggott explains which Clarion third party tools he uses, and why.

Posted Tuesday, May 24, 2005

Taboga Report's Data Exporter Library Beta 1

Beta 1 of the Taboga Report's Data Exporter Library is now available. This library is a Report Output Generator compatible library that will allow you to export your report's data (i.e. without formatting) to Excel, CSV (and other Basic format variants), and/or XML. Among other things this product writes Excel files natively, which means your users do not need to have Excel installed.

Posted Tuesday, May 24, 2005

Web Email Cloaking Utility

Ben Brady has updated the Web Email Cloaking Utility to correct a problem encountered in the generation of the Javascript encoding for email addresses. Free download.

Posted Tuesday, May 24, 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 May 25, 2005

Track	Start	Length	Size	Description
Download or stream (Free Access)	00:00:00	42:17:00	15224 K	Clarion 6.2 has been released, and Andrew and Dave talk about the new features, the subscription program, the upgrade fee, and Clarion 7.

Individual tracks				
Track 1 S	00:00:00	00:17:24	6122K	Intro, and the new Clarion 6.2 features, including TurboSQL and new templates.
Track 2 S	00:17:24	00:02:38	928K	On paying for 6.2
Track 3 S	00:20:02	00:01:30	530K	The new bug tracking system
Track 4 S	00:21:33	00:04:33	1661K	More on paying for 6.2, and the value of access to the bug fixing process
Track 5 S	00:26:16	00:06:26	2263K	The core subscription program and C7
Track 6 S	00:32:42	00:03:53	1371K	C7/Clarion.NET cross-pollination
Track 7 S	00:36:36	00:05:40	1997K	Coffee mugs, the towel incident, and Pierre Tremblay's classical guitar.

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

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 2005-05-31

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 Zauere 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.

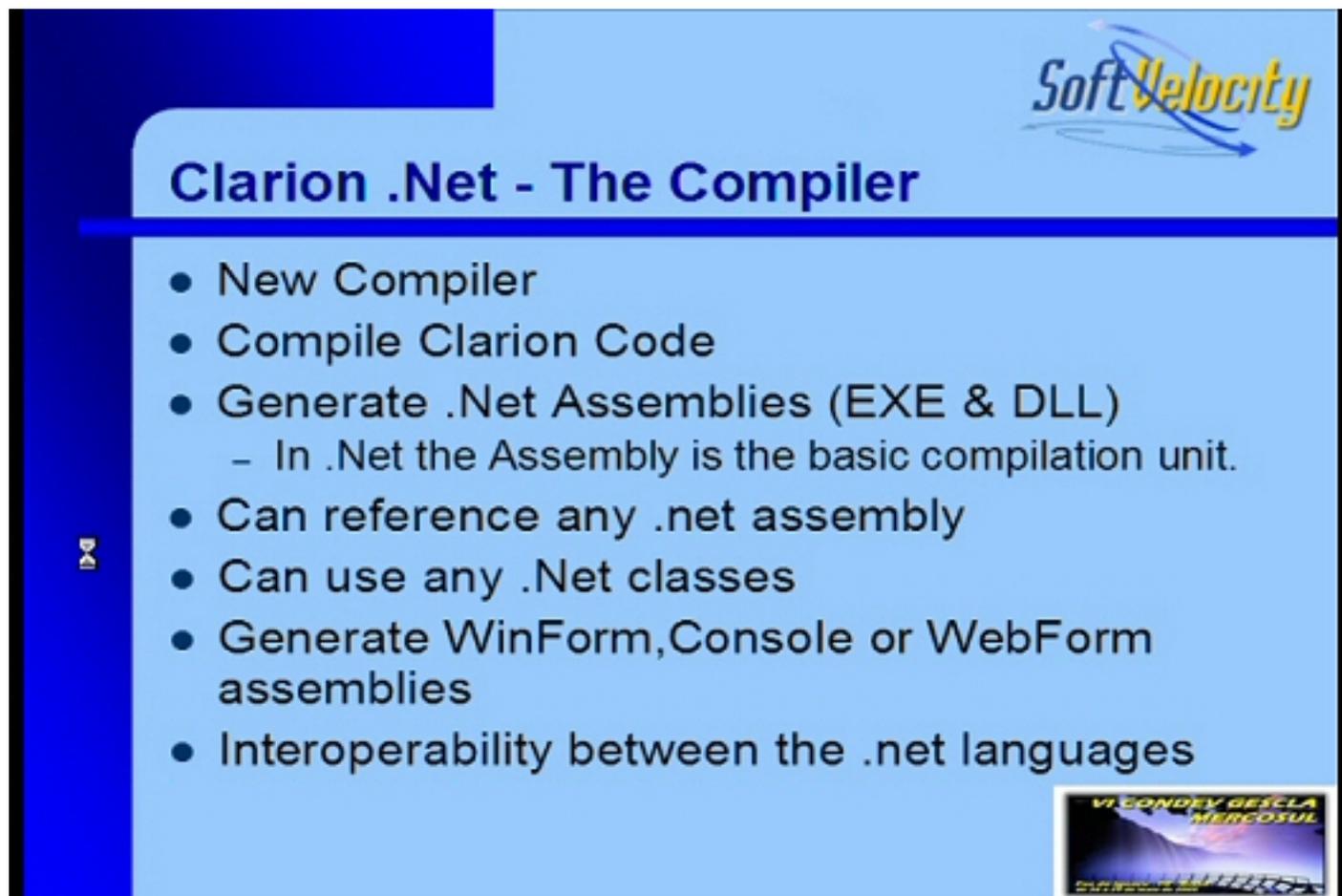
Download the AVIs

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

```

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
19

```

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

```

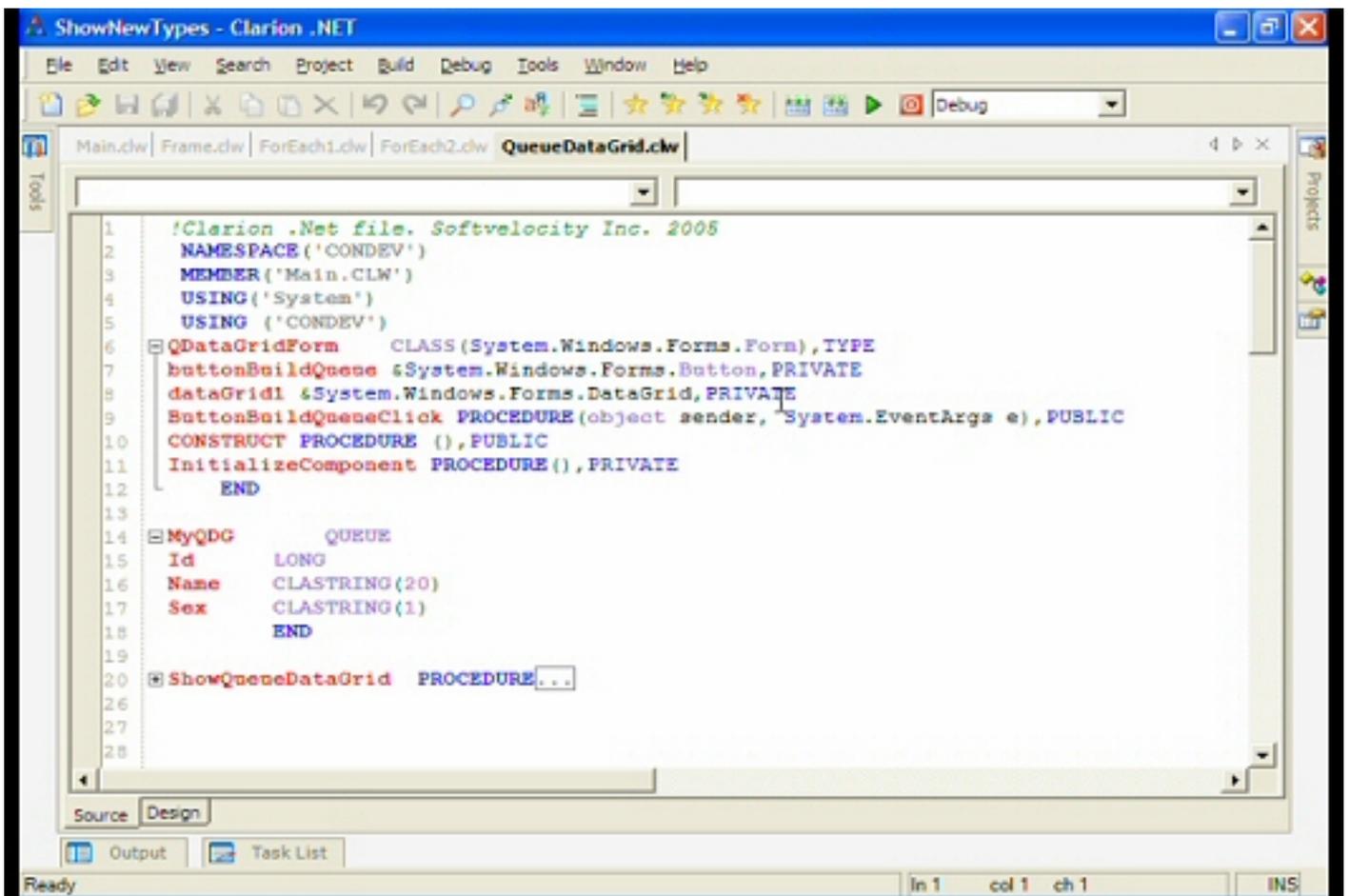
16
17 | TRY
18     IF 23 > '123'
19         LLL = 'Mayor'
20     ELSE
21         THROW new System.OutOfMemoryException()
22     END
23 CATCH (System.OutOfMemoryException)
24     LLL = 'Ocurrio una exception de memoria'
25 CATCH (System.Exception oEx)
26     LLL = 'Ocurrio una exception'
27 FINALLY
28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

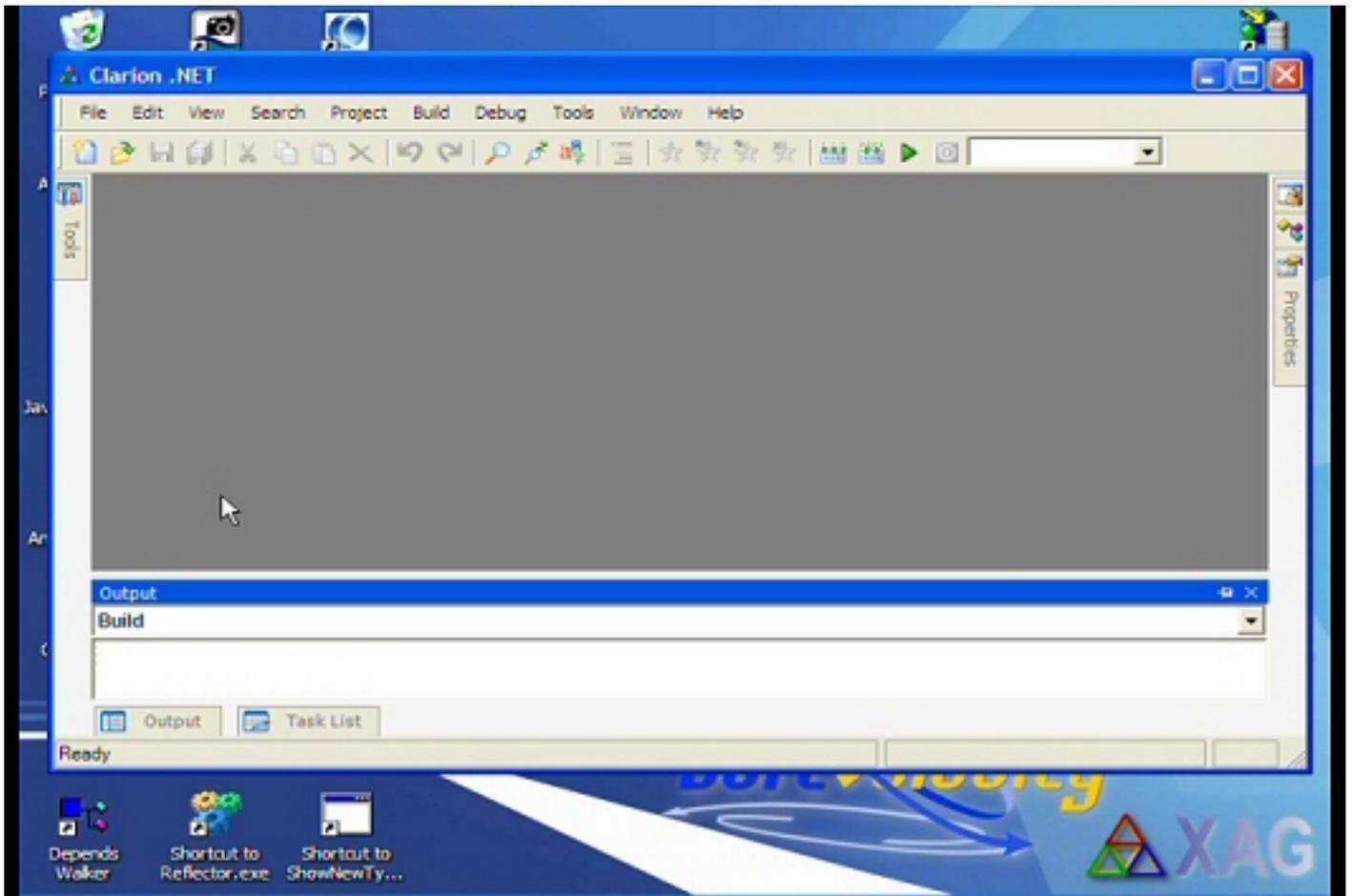
Here are a few more looks at the new IDE. I left these in for two reasons - first, I was unable to get a clean screen shot from Bob's presentation due to the higher resolution, and two, they show how configurable the IDE is. Bob had the project list on the left, while Diego has his sitting on a button 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](#).



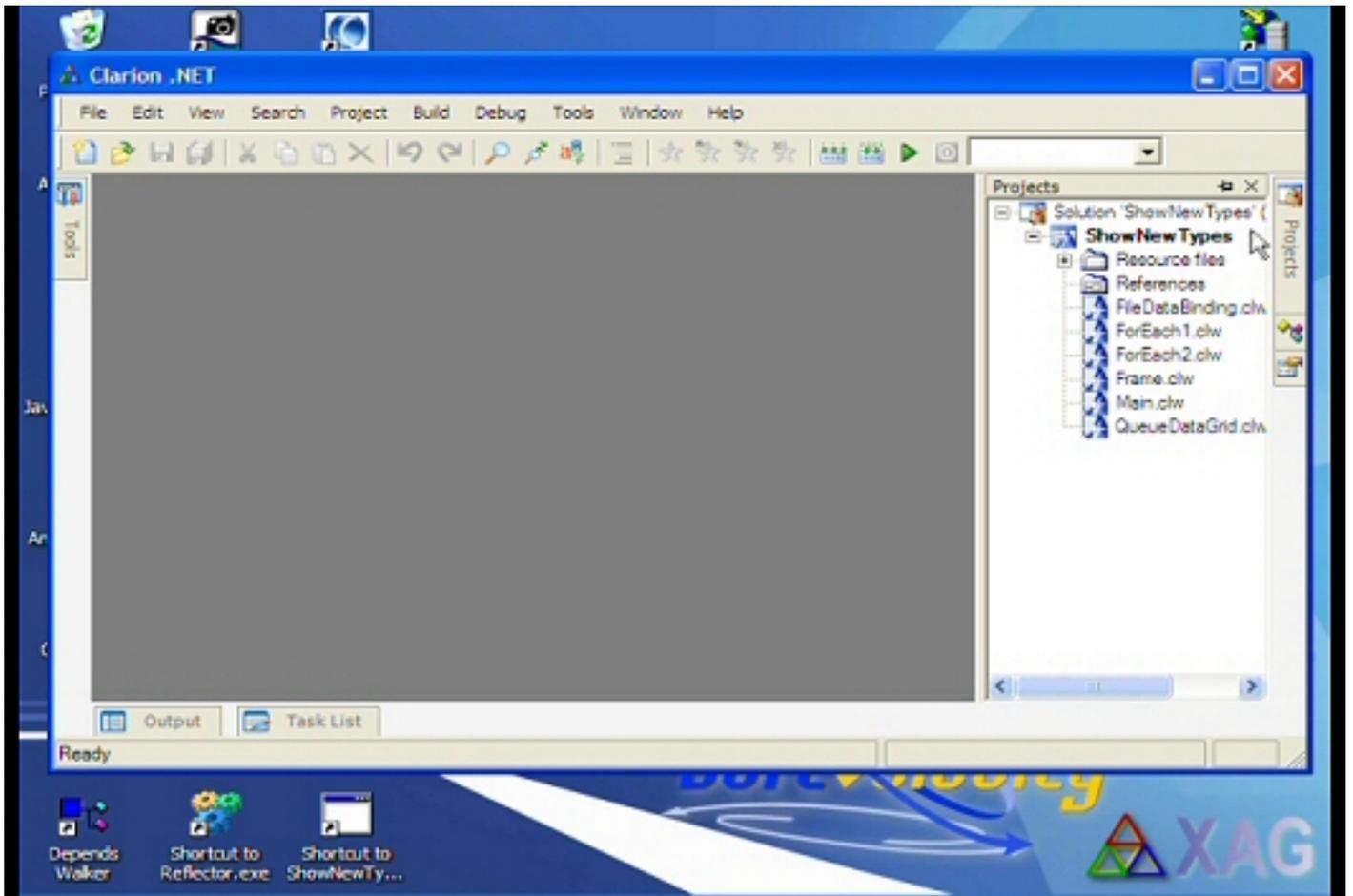
The screenshot shows the Clarion .NET IDE with the file 'QueueDataGrid.clw' open. The code is as follows:

```
1  !Clarion .Net file. Softvelocity Inc. 2005
2  NAMESPACE ('CONDEV')
3  MEMBER ('Main.CLW')
4  USING ('System')
5  USING ('CONDEV')
6  QDataGridForm CLASS (System.Windows.Forms.Form), TYPE
7  buttonBuildQueue $System.Windows.Forms.Button, PRIVATE
8  dataGrid1 $System.Windows.Forms.DataGrid, PRIVATE
9  ButtonBuildQueueClick PROCEDURE (object sender, System.EventArgs e), PUBLIC
10 CONSTRUCT PROCEDURE (), PUBLIC
11 InitializeComponent PROCEDURE (), PRIVATE
12 END
13
14 MyQDG QUEUE
15 Id LONG
16 Name CLASTRING (20)
17 Sex CLASTRING (1)
18 END
19
20 ShowQueueDataGrid PROCEDURE...
```

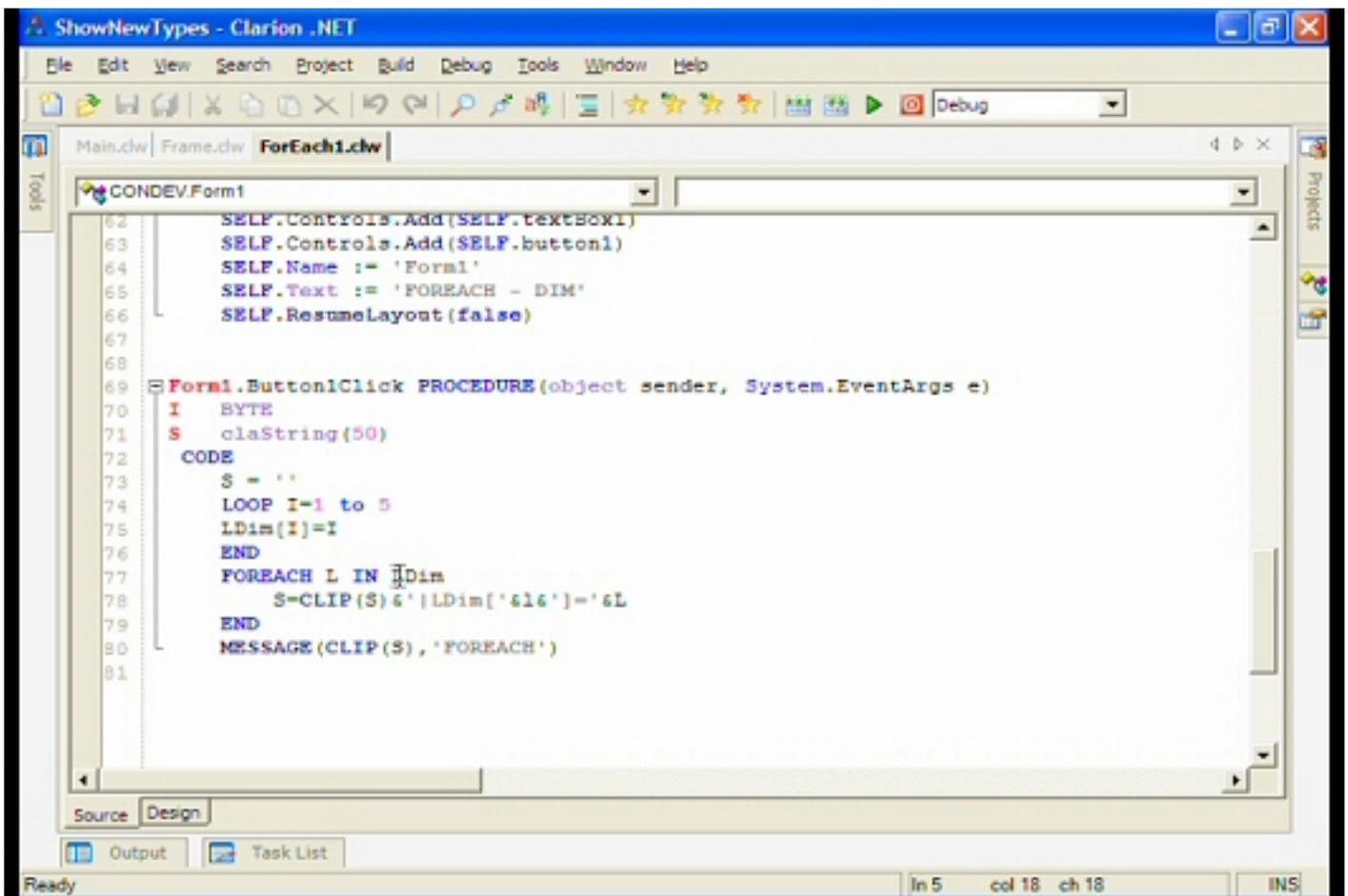
Displaying source code.



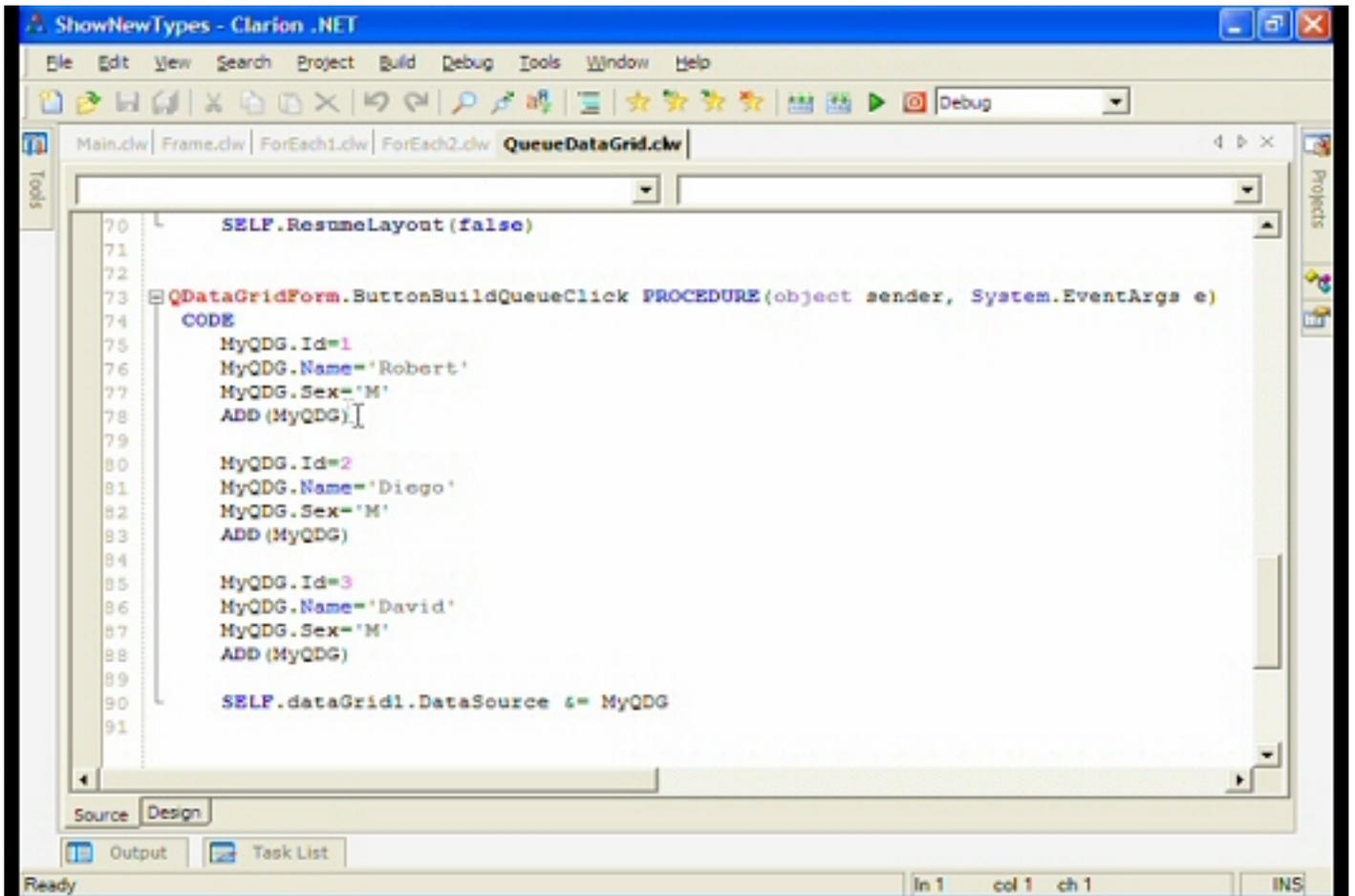
Note the docked Output pane.



Project menu on the right.

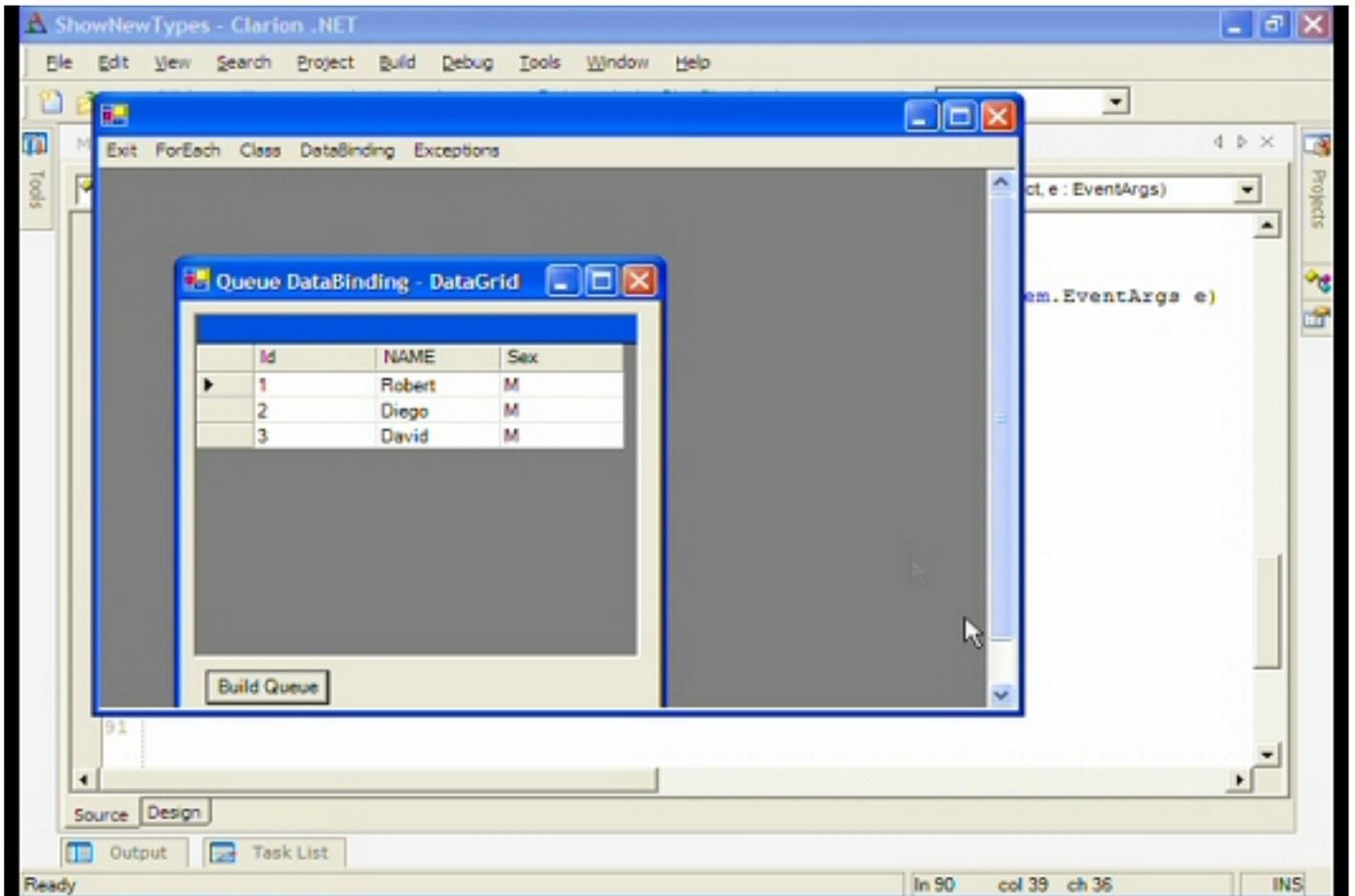


The above screen shot shows a button click handler, indicating a more mainstream approach to event handling. Clarion developers are accustomed to the `Accept` loop. Whether there is such an animal in Clarion.NET remains to be seen. This may not be a big issue since ABC developers at least are already 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.



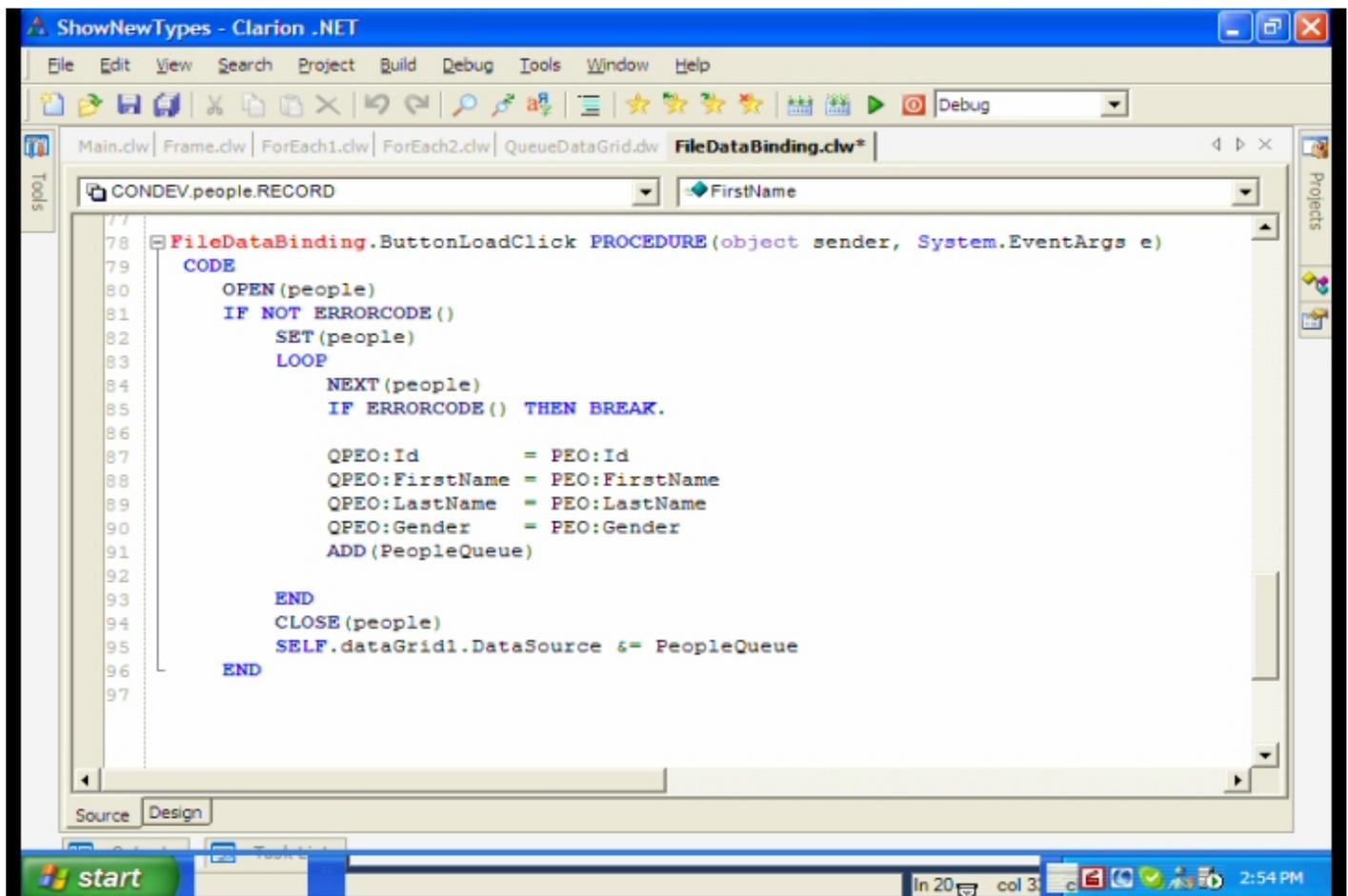
The running application with data grid. Since all of the WinForms components are available, you have a lot of choices as to how you want to present your data.

```

1  USING ('System')
2  USING ('CONDEV')
3
4  FileDataBinding CLASS(System.Windows.Forms.Form), TYPE
5
6  buttonLoad $System.Windows.Forms.Button, PRIVATE
7  dataGrid1 $System.Windows.Forms.DataGrid, PRIVATE
8  ButtonLoadClick PROCEDURE(object sender, System.EventArgs e), PUBLIC
9  CONSTRUCT PROCEDURE (), PUBLIC
10 InitializeComponent PROCEDURE (), PRIVATE
11 END
12
13
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.

Clarion Magazine

Planet Clarion Transcript: Bruce Johnson on the IP Driver

Published 2005-05-04

Dave Harms: This is Planet Clarion for December 23, 2004. I'm Dave Harms.

Andrew Guidroz: And I'm Andrew Guidroz, on the eve of Christmas Eve. Good day.

Dave Harms: Good day.

<music>

Andrew Guidroz: And today, on today's show, we're gonna have one of the people who was involved in the alphas, the betas, even in the gammas of testing the IP Driver, someone who was kind of on the inside of the know.

Dave Harms: Somebody who knows it all.

Andrew Guidroz: From all the way down. What is that? Is that Cape Horn, or Cape of Good Hope? What is that way down there at the bottom of...

Dave Harms: It's a cape of some kind. I really don't know.

Andrew Guidroz: It is a cape. I'm-- wait a minute. I'm from the US. I'm supposed to be horrible at geography. I thought we were the dumb ones, and you know, the rest of the world, you know?

Dave Harms: Yeah, yeah, but you know what? You're in between us.

Andrew Guidroz: Ah!

Dave Harms: Like when we try to look down.

Andrew Guidroz: Yeah?

Dave Harms: We try to look down to South Africa, we have to look through you, and we cannot see through your ignorance.

Andrew Guidroz: <laughs> That's what it is.

Dave Harms: So.

Andrew Guidroz: My mama was no glass maker, is that what you're trying say?

Dave Harms: <laughs> That's what I'm trying to say. Now, see? I like that one. That's good. That's very nice. So anyway, but Bruce is uh.. Bruce has an awful lot to say about the IP Driver, and he's actually going to be writing some articles for the magazine as well, which is really cool.

Andrew Guidroz: And the best part is he actually uses the thing. It's not like you and I who talk about it.

Dave Harms: Yeah, yeah. He actually knows what's going on. So uh.. let's bring-- let's bring Bruce in! Bruce?

Bruce Johnson: Okay, what we're trying to do is get you to talk one at a time, because when you both speak, it breaks up completely.

Dave Harms: Oh, that's gonna be trouble!

Andrew Guidroz: Oh, boy! We always...

Bruce Johnson: We're gonna ruin your whole, your sort of banter thing going backwards and forwards there. You know? But...

Dave Harms: Well, only...

Andrew Guidroz: Sort of banter? I thought we had achieved banter.

Dave Harms: <laughs>

Andrew Guidroz: Only it's...

Bruce Johnson: I guess it's different to have fun, but then when you both speak, it kind of mixes the two together and makes it real difficult to tell the one from the other.

Dave Harms: Yeah, well, only as far as you're concerned, because I can- I can get both of us when-- Andrew and I can talk over top of each other, and I still get us both. But I think if you guys talk at the same time, it'll probably be the same problem for me. I won't be able to hear you both.

Bruce Johnson: Yeah, what happens is 'cause the sound quality is very choppy. So it kind of goes on and off quite quickly. And when one's talking I can follow it okay. But when you both speak, then it's almost impossible to figure out what's going on. So there ya go!

Andrew Guidroz: Would it be better if we switched to your native language?

Dave Harms: <laughs>

Bruce Johnson: <laughs> Well, it would be, but it would have about the same effect of both speaking together, in that I wouldn't have a clue what you were saying.

Andrew Guidroz: <laughs>

Andrew Guidroz: What is your native language, Bruce?

Bruce Johnson: I'm English. But we have eleven. So, oh here we go.

Dave Harms: How many languages do you speak?

Bruce Johnson: Oh, I speak two. Only two.

Dave Harms: Which are? English we presume is one of them, but...

Bruce Johnson: <laughs> Some would say yes, some would say no. English is one, Afrikaans is the other.

Dave Harms: Afrikaans, yeah.

Bruce Johnson: Afrikaans is a sort of...

Dave Harms: A Dutch kind of blend thing, right?

Bruce Johnson: Yeah, it's a Germanic language, but it's primary influence is Dutch. The closest you'd come today it would be Flemish. Flemish as spoken in Southern Holland is very, very close to Afrikaans.

Dave Harms: Huh, well, there ya go. See I...

Bruce Johnson: Dutch, I can understand, but it's quite a different dialect. Well, it's more than a dialect. It's a different language.

Andrew Guidroz: I like the one that James Cook speaks from time to time. The one with all the clicking in it.

Dave Harms: <laughs>

Andrew Guidroz: Do you know what I'm talking about?

Dave Harms: You're talking "The Gods Must be Crazy."

Andrew Guidroz: Well, it's similar to that one.

Bruce Johnson: Oh, yeah! Okay. That's...

Andrew Guidroz: Now is that...

Dave Harms: Yeah, was that a rea-- I thought when I listened to that, "That's not real. They're just doing it with sound effects."

Andrew Guidroz: No, but James Cook, there is a language in South Africa. Do you know what it's called, Bruce, that has a lot more of the <clicking sound> kind of stuff going on.

Bruce Johnson: Yeah, okay, that style, having clicks in the language, that's very common for African languages. The one James Cook speaks, which is Xhosa, is one of the bigger languages here. But that's nothing like what the bushmen speak, which is what you would've heard in "Gods Must be Crazy." And that's a genuine language.

Dave Harms: So that was real. That was real.

Bruce Johnson: That's real. Oh, yes, very much so.

Dave Harms: Wow!

Bruce Johnson: That whole culture is very real. Still real today, although being more kind of Westernized now. And influenced...

Dave Harms: Well, they've all been to the movies now, I guess. <laughs>

Bruce Johnson: <laughs> Well, you know, like we have these sort of cultural tourism places where you can go, and you can see people kind of behaving like they did, you know, a thousand years ago. Or two or three hundred years ago, maybe.

Dave Harms: Well, we have that here, too. It's called a pub.

Bruce Johnson: <laughs>

Andrew Guidroz: Not to be a party-pooper, but maybe we should cut to the chase, considering everything going on.

Dave Harms: Andrew's in the middle of a massive political race, and he's the one doing most of the racing, it sounds like. So...

Andrew Guidroz: Yes.

Dave Harms: It's very hard-- yeah, he's a hard guy to reach these days.

Andrew Guidroz: Here is what I want to ask. I have noticed some talking about the IP Driver up on the newsgroups. And a couple of people have basically taken a standard Clarion application with form and browse, and attempted to use it across the Internet, using the IP Driver. Some people have reported great success, who basically are in a situation where they've got the same ISP for each location; they get pretty quick response time, because they're only a couple of hops away. Those people who go through multiple ISPs seem to complain a lot about response times so poor that it just isn't usable.

Dave Harms: Yeah, 20-second stuff save a record. 20-seconds to load a browse, stuff like that.

Andrew Guidroz: Yeah, so, what's your gut, Bruce? What do you think the IP Driver's all about?

Bruce Johnson: I think there's a-- it's been taken out of context a little bit. You guys were chatting in the podcast about some of the things that Bob Foreman said at Devcon. And if you go back and listen to Bob, there're very much things that you really could do better just with a normal XML file, or a normal web client type thing.

Andrew Guidroz: Using pure text.

Bruce Johnson: If I was-- yeah, I mean, if I was gonna do a version updating system, I wouldn't use IP Driver, I'd put-- we do-- we've got several products that do it.

Andrew Guidroz: You'd use that

Bruce Johnson: You'd put a little file on your web server, and you'd fetch it with a web client, you know, HTTP and you'd just fetch the thing. It's all of four bytes. If you need to then get a download, you'd use a web client and you fetch the install file. So using IP Driver for that sort of thing, even a job ro-- you know, he talked about a job robot, which is essentially like a remote procedure call, you might call it. And there're very easy ways to do that. You could either make up a protocol, you could use SOAP, or anything of that nature. So I wouldn't be using an IP Driver for that sort of thing. And I think people got a little bit kind of sidetracked down that road. In my opinion, it's there when you have data. It's for accessing data. Normally, it's a client server against a set of data tables.

Andrew Guidroz: But wait. The problem that I'm having is browse and forms are looking up data also.

Bruce Johnson: Absolutely.

Andrew Guidroz: If you...

Bruce Johnson: Let's go forward. So a browse and form is an ideal use for it. In terms of whether you use it on a WAN or not, and obviously that includes the Internet-- I'm gonna tell you stuff you know-- but it's gonna matter is what your latency is and what your throughput is. So...

Dave Harms: I'm kind of curious about, and I don't know if you've had a chance to

investigate this, Bruce, but I mean, I do remote database access everyday, because the Clarion Magazine web server's in Texas, and I'm in Winnipeg, and we're-- what is that? That's like-- that's a couple thousand miles apart.

Andrew Guidroz: At least a six six-pack drive.

Dave Harms: Yeah. And... if you're from Louisiana. And I have reasonably good response times, you know, using a MySQL on an encrypted connection. What I don't understand is why, with the IP Driver, I couldn't expect similar performance. Do you have any idea why that is?

Bruce Johnson: I think you would get in the same order of magnitude similar performance. I haven't, you know, one would have to set up some fairly sophisticated tests to see how close it actually came. I think the IP Driver could improve, but-- and probably would improve given that it's a 1.0 release.

Andrew Guidroz: Is it the IP Driver that needs the improvement, or is it actually the class is accessing the IP Driver?

Bruce Johnson: No, well, there are no classes accessing the IP Driver.

Dave Harms: Well, there's just...

Andrew Guidroz: It's just a driver.

Bruce Johnson: It's just a driver.

Andrew Guidroz: Right, but there's ABC in the middle of this, and ABC may be non-optimal for a WAN type situation in this case with long latency.

Bruce Johnson: That's true in some cases. One has to be real careful about judging ABC as this kind of nebulous thing. There's a huge disparity in people's opinion about how good is ABC is or isn't with SQL. It would be no different to SQL, and perhaps the fairest measure if you wanted to do that would be saying take a browse and point it at a database, you know, a remote database, and take a browse and point it at a remote IP site. If you're comparing apples to apples in that sense, then I think you'll find performance is similar. I wouldn't really know how close it would get. I'm guessing something like MySQL has had a bunch of time and effort spent on optimizing that connection. And I suspect things, you know, you've gotta worry about when're you going to start compressing your

packets? Almost certainly, something like MySQL does packet compression.

Andrew Guidroz: There's some people out there that have basically tried to use MySQL to go to certain projects, and like a lot of folks, they hit the wall. You know, when you get to that certain area where you know a good bit about SQL, but you don't know everything, then you hit the wall. Then what they start to find is, "Hey, this IP Driver's coming, maybe I don't have to, you know, solve MySQL problems, maybe all I have to do is wait for this IP Driver to show up, flip the switch."

Dave Harms: Yeah, although they've actually-- Bob, when he was in here was very clear-- I think Bob Foreman, too, at Devcon that the IP Driver, one thing that it is definitely not intended is to be a replacement for SQL.

Andrew Guidroz: Right.

Dave Harms: I would say that it's intended, and it did make the comment at Devcon that it's intended, certainly one of the applications is on a local area network, when you're-- it gets rid of the op locks issues. Because if you have a server that is...

Andrew Guidroz: Because of the one server application is taking care of all the locking situations.

Dave Harms: The one server application takes care of all the locking, so you don't have different applications fighting for locks on the files. So if you're facing data corruption on a local area network, the IP Driver would certainly be one way to deal with that issue.

Bruce Johnson: Yes, I would say that the killer feature for it is definitely the LAN more than the WAN. There's been a lot of talk about the WAN on the Internet. At the end of the day, whether you get better performance, or less performance in SQL, you will get a hell of a lot better performance than using TopSpeed files, for example, which is just plain impossible. On a LAN, on the other hand, is where the thing really steps up to the plate in my opinion. File corruptions just completely go away. Now file corruptions are still very much a problem with your basic rollouts. We've got excess of 2,000 sites running our T&A system.

Andrew Guidroz: Your what system?

Dave Harms: Yeah, your what system?

Andrew Guidroz: Your what system?

Bruce Johnson: Yeah, yeah, okay. Time and Attendance.

Andrew Guidroz: Time and Attendance.

Bruce Johnson: I know you're North Americans, you know.

Dave Harms: We're just thinking about this...

Bruce Johnson: One track.

Dave Harms: We're just thinking if this was sort of the companion to the Hot Dates product.

Bruce Johnson: Yeah, that's it. You know, and it's a TopSpeed system. It's a TPS-based system. But guys'll still be doing silly things. They'll still have networks set up wrong. And which can lead to file corruptions. You turn a workstation off with a plug, you get a file corruption. Or you've certainly got a really good chance of getting one, put it that way. So, in terms of moving on a LAN, you're moving from a shared file system, to a client server system. Not SQL, but client server. There's a difference between those two. But that's essentially what you're doing on a LAN.

Andrew Guidroz: And you can also run it on a local machine, both the client and the server, correct?

Bruce Johnson: Oh, yeah, yes, absolutely.

Andrew Guidroz: So although...

Dave Harms: But there's no apparent-- there's no real benefit to doing that. Or is there?

Andrew Guidroz: Well, the benefit is you could write a single app, that'll run either on a single workstation, or it'll work in a LAN environment.

Bruce Johnson: Yeah, I mean, there are techniques for making a single program run as an IP version, or a native TopSpeed version. We're doing that at the moment using things like [Multi-Proj](#) and so on. In fact that's the whole test cycle I went through with IP Driver. The first thing I do is I build an IP Driver version. And that means I can test that in complete

isolation from my real kind of working out [unintelligible]. Gives me a kind of an easy changeover. There are definitely places where the TopSpeed Driver version will still exist. If I'm gonna build an app for a single machine, TopSpeed is, it just doesn't get any faster. Except for some sort of backend stuff, where SQL can do, you know, change to thousand records in one go, apart from that kind of thing.

Dave Harms: All right. Is there a risk? Let me put it this way. Is there a bit of a risk here that people are getting all excited about the use of the IP Driver in an Internet scenario, in a WAN scenario, and they're completely missing the whole point of it. And they're using it for something that it's not designed for, and it is gonna cause a lot of disappointment and discouragement.

Bruce Johnson: [Pause] Yes.

Dave Harms: In a nutshell.

Bruce Johnson: In a nutshell. I certainly-- and I can only really speak for myself-- we've got, as I said, this application, and I see absolute-- we've got guys crying out for the LAN version of an IP Driver enabled system. Bear in mind that a lot of LANs have got serious bandwidth problems. We talk about 100-megabit LANs as if, you know, well, that's great. I mean, our office, it is great. But we've got clients where...

Andrew Guidroz: But it doesn't work when you've got dozens of clients on a small network, it just doesn't hold up.

Bruce Johnson: Yeah, I mean, you wind the user up a bit, the guys toss in one or two radio links just for kicks, before you know it, the network is...

Andrew Guidroz: He's watching live video of a [clucking chicken](#) on the Internet, or something like that.

Bruce Johnson: Yeah. I mean, you know, we go to some big sites and so on, and the guys'll say, "Ah, you know, it performs really slowly." And we'll run some basic, basic sort of benchmarks across the network, just reading and writing tables. And sometimes stuff gets...

Andrew Guidroz: It just falls over.

Bruce Johnson: Yeah, I mean, here it's taking maybe a minute to do a process. It could

be taking 30 or 40 minutes there. And that's down to network bandwidth. Now, what'll happen is that if you can-- if you start using something like an IP Driver, the amount of bandwidth drops off the chart. I've been doing testing, and just using sort of Windows XP monitors and so on. The amount of network bandwidth [used] goes right, right down.

Andrew Guidroz: How quickly could you change an application, say, that has TPS files? You know, using the traditional native driver, if you wanted to go to IP Driver?

Bruce Johnson: Quick. Yeah, I mean, one of the thing we worked on when, you know, building up the whole IP Driver thing, when I got involved, one of the big things was how long does it take? It's pretty darn quick, you know? It depends on the size of your application a little bit. If you've got a 40-app suite, it's gonna take a little bit longer. But if you want to pick up a single app program, it'll take you-- once you've kind of done it a couple of times-- you've got a bit of the hang of it-- probably less than five minutes. To turn it off is about three seconds, 'cause it's got a switch. So you can apply it to your TopSpeed application, and then you can turn it off, if you don't like it.

Andrew Guidroz: One switch.

Bruce Johnson: Yeah.

Andrew Guidroz: So basically once you set it up to say, "This is IP Driver compatible," then there is the famous SoftVelocity checkbox that says, "Turn it on," or "Turn it off."

Bruce Johnson: That's it.

Andrew Guidroz: Yeah.

Bruce Johnson: That's it.

Andrew Guidroz: That's cool!

Dave Harms: What about the whole server side filtering issue? Are there things there that need to be taken advantage of?

Bruce Johnson: The trick with server side filtering is exactly the same with IP as with SQL. Is that you've gotta do your filter in such a way that it can be done on the server side. The moment you include something in your filter that can only be done on the client side, for example, a bound function. You know, a function that you bound, and so on, then

the whole filter, including the range has to come down to the client. Now what'll happen is you'll open a browse, and it'll take minutes to display. Ages and ages. And the best thing to do then is just go turn off the range and filter and see what happens. And typically it comes up real quick. So you put the range and filter back one piece at a time, until you find the piece that's- that's not server side compatible.

Dave Harms: That's really the only scenario in which the IP Driver would, on a LAN, the IP Driver might be slower than native TPS access. Is that right?

Bruce Johnson: Yeah, I guess. I'd have to think about that...

Dave Harms: Or would the native TPS be just as slow?

Andrew Guidroz: No, no, no.

Dave Harms: Would the native TPS be just as slow?

Bruce Johnson: No, well, the native TPS is...

Andrew Guidroz: Well, it would still be faster. It would still be faster, because even if you're getting records that need to be filtered, you're getting just those records. You're not getting the block of data that you'd be getting by using the TPS driver across the network. So you're not getting the whole disk IO push and pull of a traditional ISAM file. You're getting just the record.

Bruce Johnson: Yeah.

Andrew Guidroz: So it would still be faster. I mean, it's just...

Bruce Johnson: I'd agree it's probably similar.

Andrew Guidroz: As long as you're across the network.

Bruce Johnson: Yeah.

Andrew Guidroz: Locally, certain.

Bruce Johnson: I know, but that's...

Dave Harms: And not a major point probably.

Bruce Johnson: No, and in fact we did a page when I was doing the SQL stuff, so there's one of our "tip of the week" things on our website mentions that when you're moving to SQL, you have exactly the same issue. And all you do is you just look at the code, and you move your client side filtering into your embed point. Into the validate record embed point, and you're done. And that has no impact on the TopSpeed version. So your TopSpeed version is still alive and well, you haven't broken it. If you did want to do specific code, there are some functions where you've got to think a little bit carefully about whether this is supposed to happen on the client or on the server. I'll give you an example, it's if-- let's say you're doing a file import, and you've got your accounting application, or your payroll or whatever, and you're going to import a text file. Now that import file is typically a local file, not an IP Driver file. And you can tell the system that. You can say, "No, no, this file is local, it's not an IP file." In other cases you may say, "No, but the file is going to appear on the server side." So things like file dialogue may or may not kind of apply. Those are the little corners of the app that you need to think about. I came across having two in my system, where you've just gotta think a little bit and say, "Hang on. What is the goal here? Where's the file actually gonna be? Is it gonna be on the client's PC, 'cause the guy just got it on a floppy disk, or in his email or something? Or is it gonna be loaded onto the server? Is it actually going to be dropped onto the server by another system?" Those are kind of minor areas. And there's a project define, which is set by the client template, the IP Driver client template, set to define on and off, so that you can put conditional code in your program to handle that, and say, "Okay, if I'm compiling this with the IP Driver on, I want you to do it this way. If I'm compiling this with the IP Driver off, I want you to do it that way."

Andrew Guidroz: Okay, so let's cut the whole thing down to the bone, okay? If you're using SQL currently, there really isn't a big incentive to go from the SQL to the IP Driver, because you may be losing performance, right?

Bruce Johnson: Yeah, I would say if you're using SQL, there's... there's very little... there was one comment on the newsgroups about security and so on. You're kind of effectively hiding base SQL away. But...

Andrew Guidroz: Now, from there, if you're using TPS files, this definitely is going to be a big boom in your speed when you're dealing across a LAN. But you're mileage may...

Bruce Johnson: Well, stop there. No, no, no. I don't think I-- in my testing I haven't seen the IP Driver go real-- go much quicker over a LAN.

Dave Harms: But it takes less bandwidth.

Bruce Johnson: On my LAN. But my LAN-- yeah, but my LAN is empty all the time.

Andrew Guidroz: Ah, because yours is already optimized, so you don't see a big difference. Now if you would be...

Bruce Johnson: Well, it's not so much that. It's that the LAN has got bandwidth to spare. So I can run the TopSpeed files flat out, and it uses 25 percent of the bandwidth. But I get all the bandwidth I can use. So it's-- the bandwidth is not a bottleneck.

Dave Harms: And what's the percentage of band-- what's the reduction in bandwidth usage, ballpark? What would you say? What's the range of reduction?

Bruce Johnson: It's gonna be different for different applications in different places. And one of the tests I did I dropped it down from about 12.5 percent network usage, to about two percent network usage.

Andrew Guidroz: Zow!

Dave Harms: That's pretty significant! Yeah.

Bruce Johnson: That's a lot.

Andrew Guidroz: And to me...

Bruce Johnson: And bear in mind that the way Ethernet works, it's not a question that you get up to 100 percent. The more full it is, the more the performance degrades, 'cause you get more collisions.

Andrew Guidroz: Right.

Bruce Johnson: So typically, any network that's running over about 50 percent is actually in trouble. You're gonna have more collisions, and that creates more bandwidth, which kind of snowballs on itself. So you wanna keep that bandwidth low. How much of a speed increase you're gonna get depends an awful lot on how busy is the network already. On a machine, on a LAN connection, the speeds are close enough that you can't really tell the difference. If you sit there with a stopwatch and you measure it, there's about a two to

three percent difference, sometimes a little bit more, sometimes a little bit less. Bear in mind the IP does extra work. There's an extra layer involved. So there's a tradeoff. But certainly for LAN usage, if you're using a TopSpeed file today, you're gonna get two major benefits. The one is reliability is gonna go through the ceiling. And that's no small thing, if you're trying to support multiple applications. If you don't have control over the networks they're going onto, that's not small thing. And the other thing is it's gonna use less bandwidth. And especially in bigger places, guys will get very upset with you if you start using a lot of bandwidth. Especially if the networks are clogged. And then you will get performance...

Dave Harms: Let's maybe reposition this a little bit and say, you know, instead of the IP Driver, start thinking about it as the TopSpeed Driver performance upgrade.

Andrew Guidroz: Yeah? Or...

Bruce Johnson: Yeah, it's client server. It's the step from sharing files to client server.

Dave Harms: It's the client server...

Bruce Johnson: Without going to full-blown SQL.

Dave Harms: It's the client server version of the TPS Driver is really what it is.

Bruce Johnson: That's exactly what it is.

Andrew Guidroz: There's a third scenario in this thing that we haven't discussed either. It's legacy data, and the fact that SQL, of course, is great across the Internet-- won't say it's great-- it's the solution across the Internet. But certainly, if you have legacy data already setting in a TPS file, and you're trying to get it out to remote users, this is certainly one more solution for you, that before your TPS files were basically hard to get at for you, if you live anywhere around the world. And using this now, you've got one more way to go and look up a record. You may have-- to me, you could write an application and I guess it would function a bit like a web browser in that it would be able to handle latency a lot better. But you could go out and look up a TPS record, if you so desire.

Dave Harms: Or you could do a background process and be grabbing data from a web site in a very easy manner, and that's something that a user might later deal with offline.

Andrew Guidroz: So, in essence there would almost need to be a modified browse form

paradigm to act more like you're in a long latency environment. Right? Does that make sense?

Bruce Johnson: Yeah, we still-- I wouldn't write off the Internet stuff just yet.

Andrew Guidroz: Okay.

Bruce Johnson: I've been testing it over an ISDN dial-up, a 128k ISDN dial-up, and I wouldn't say it's terrible. TPS performance over that, for example, is just completely non-existent. The stuff we were doing was reasonably, it was usable. I wouldn't say it's quick, especially for sort of those who're used to LAN speeds. But it's definitely usable. And I think there it was also, you know, perhaps people were very ambitious in saying, "Let me take my app as it is today, and try and kind of bang it across an Internet connection." Even if you're gonna do SQL across an Internet connection, you're gonna think long and hard about the fields that you're reading.

Andrew Guidroz: Sure.

Bruce Johnson: What is all included in your browse? You know, you can do very fancy things basically for nothing on a LAN. You can do look-ups all over the place very easy on creative stuff. On a WAN, you'd get a lot more conservative and bring those fields right down. You would optimize what you're fetching, you'd make sure that a 2k memo is definitely not coming down, that kind of thing. I think guys were really excited and saying, "Hey, this is gonna give me TopSpeed performance across a wide area network." And that's not gonna happen. The difference, I guess, in terms of, yes, you've now got a TopSpeed app, is that you do get client server in a couple of minutes. And the ability to go back with the flip of the switch, which is a step along the road to SQL.

Dave Harms: Uhm hm. And it's still surprising to me the number of people who have not made the jump to SQL. But there's a lot involved in making that jump. So there's some huge benefits here.

Andrew Guidroz: The word is legacy. You've already got the legacy worth of commitment to it. And you've got those handful of clients that the TPS thing does work well for a small network with a-- I won't say a handful-- but a certain number of users. You know, there're a lot of times the old TPS thing works. And I've got a bunch of clients it works perfectly for, you know?

Bruce Johnson: Yeah.

Andrew Guidroz: And do I really want to go through the effort of-- does the customer want to spend the money to reengineer it for SQL? Probably not, you know? They're satisfied with what they have today.

Bruce Johnson: Yeah, well, we have the same thing with T&A. We have a SQL version and a TopSpeed version. The TopSpeed version will sell, it's the bread and butter, and I've always sold more copies, probably nine to one. And that's just because SQL has for all its benefits, it has some disadvantages when it comes to rolling out hundreds or thousands of copies. There's a maintenance issue. There's an installation issue. There's all those kind of things. If I ask a client to send me your data files, that's an absolutely trivial thing to do with TopSpeed. It's not so trivial with SQL. So, SQL has its place. It's absolutely, it's got a great place. But it's not for every application in every situation. The IP Driver fits very, very nicely. It's not, I think, one shouldn't try and see it like as the solver of the world's problems. It's not a replacement for SQL. I don't think it's a replacement for HTTP either. And a lot of things people have talked about, you can do better with HTTP. And stop messing around. But it does serve an incredibly useful purpose, and it has a very, very big place. When you couple it with a couple of other technologies, the Dyna Driver, and Memory driver, then things start to get a little more interesting again. Because you can do some really funky stuff. There's nothing to stop you caching a file locally as a TPS file. Which you'd fetch and synchronize with a file, an IP file over the WAN. So things-- if you look at your average dictionary, there's a huge number of tables which either never change, or change very seldomly. Those files and those records don't need to come down the pipe all the time. They can be fetched once, and they can be maintained after that, so that you go and fetch them again, let's say, in a month's time, or whenever the file changed. You've got a list of zip codes, you've got a list of departments, you've got a list of branches, these things don't change. And there's no reason to go and fetch that data across a WAN, be it an SQL backend, or an IP backend, or whatever. So using some of the other-- well, using the drivers together, not only the IP Driver-- but you could do it again as SQL. You can create local caches. Within Memory Driver, you can create a local cache in memory of a TopSpeed file that's on a disk, or of a TopSpeed that is in itself a cache of a remote file. So each driver has a place, but when you start putting them together, you can start doing some really funky things. If bandwidth is your big bottleneck, you can do some really impressive things to bring your bandwidth down.

Andrew Guidroz: Cool.

Bruce Johnson: When you start mixing them together. That's a little bit down the road, I guess, when we gotta wait for the Dyna Driver to come out and see how it all fits together,

but if you start putting them together creatively, then there is a place for that. But let's go back just to the IP Driver. I think it's for guys who're running flat file systems at the moment. It turns them into client server systems. It's why a lot of guys went to Btrieve from TopSpeed.

Andrew Guidroz: Right.

Bruce Johnson: It was for the client server aspects of Btrieve, and that's exactly the same reasons it'll drive you to using an IP Driver for your TopSpeed files. Guys are talking about ISPs hosting the, you know, the server side. And it's-- I don't know. That's not for me. If I'm putting a server in, it's going in at the site where it's gonna get used. I'm not looking to stop in a...

Dave Harms: Right.

Andrew Guidroz: Right.

Bruce Johnson: ISP set up where...

Andrew Guidroz: But really...

Dave Harms: And really...

Andrew Guidroz: But wait. What you have to also remember, though, is that certain organizations have more than one location. So yeah, you're not gonna host it, but certainly the flip side is that if you've got an organization with, say, ten locations, one side has to host it, you know?

Dave Harms: Yeah. And you know one thing...

Bruce Johnson: Well, one or more. One or more. We're looking at adding some replication functionality to it, so that your servers, you could have three servers going.

Dave Harms: Where would a guy get [replication functionality](#), I wonder?

Andrew Guidroz: Yeah, where would you get replication?

Bruce Johnson: Yeah, I don't know where you get that.

Andrew Guidroz: <laughs>

Bruce Johnson: I mean, but I'm sure someone out there...

Dave Harms: I'm sure somebody has it. I'm sure somebody has it.

Bruce Johnson: Would come out with it, yeah.

Andrew Guidroz: Well, Bruce, we want to thank you a whole bunch for all the time that you spent with us today. And it's good to talk to somebody-- you are actually involved with helping SoftVelocity test this thing, and you run it through a lot of its paces.

Dave Harms: And yeah, I think you've opened our eyes to a few things that we hadn't really realized. So thank you very much!

Bruce Johnson: That's my pleasure. Anytime!

Dave Harms: All right! We'll be talking to you again, I'm sure.

Dave Harms: Hey, I've got one thing I wanna just mention really briefly. This is by way of follow-up to our previous pod cast where talked about color theory and all that kind of stuff.

Andrew Guidroz: Right, uhm hm.

Dave Harms: And there were a couple of comments, conversation going on in the newsgroups, and a couple of things I wanted to mention.

Andrew Guidroz: We haven't mentioned this before, have we? Color Wheel Pro? Did we talk about that? I don't think we did.

Dave Harms: No, we haven't. It's excellent.

Andrew Guidroz: The little bit of...

Dave Harms: Excellent little thing. Yeah, just go to www.color-wheel-pro.com and there's a utility there that will let you, you know, use a color wheel to do some web site designer application design. It just kind of helps you with the different color things. And that was something Jim Kane pointed out. And Charles Edmonds, his favorite tool for that

kind of work was something called Color Impact, which is from TigerColor.com. And one other thing that I wanted to mention was something called the Web Smart Palette. So you might have heard about the Web Safe Palette, which is 256 colors, it's safe to use, that would display properly in any browser, I guess, because the idea being that maybe some people only have, what, 8-bit color, and you don't wanna have your website looking really ugly. These days, of course, you can pretty much count on at least 16-bit color, and so you're not quite as limited. Anyway, there's-- somebody's set up a web site and suggested a Web Smart Palette, and this is at MoreCrayons.com.

Andrew Guidroz: MoreCrayons, pretty name.

Dave Harms: MoreCrayons, yeah. Subtitled "Colors for Web Designers." Yeah, you know what I mean? It's actually a 216, the Color Browser Safe Palette was really only 216 colors.

Andrew Guidroz: I didn't know that. That's-- anyway, there ya go.

Dave Harms: Yeah.

Andrew Guidroz: Also my desktop is totally locked up. So the odds that we continue having this conversation for a long period of time...

Dave Harms: I can hear you fine. I can hear you fine.

Andrew Guidroz: Well, the odds are slim to none that this will continue for a long period.

Dave Harms: Okay. Okay, well, I guess we better wrap this up.

Andrew Guidroz: And I'm Andrew Guidroz.

Dave Harms: And I'm Dave Harms. Good day.

Andrew Guidroz: Good day!

Reader Comments

[Add a comment](#)

Clarion Magazine

Modifying the Common Dialogs: How Hard Could It Be?

by Jim Kane

Published 2005-05-12

From time to time I've thought it would be nice to add a button or other control to one of the built-in common dialogs for opening a file, saving a file, or printing. In my current project I thought it would be nice to add a multi-page TIFF preview button to the file open dialog (see Figure 1, below). How hard could it be?

Well some 12 hours later, bleary eyed and frustrated, I had not made a button appear, but at least I knew how to do it. I also was getting a feeling for how hard it could be!

The trail to eventual success started by looking in [MSDN](#) at the API procedure (`GetOpenFileName()`) for opening a file. It only takes one parameter, a group structure called [OpenFileName](#). I followed the link in MSDN to the description of the `OpenFileName` structure and searched the field by field descriptions for the way to customize the dialog. The `lpTemplateName` member held the most promise. It said all I had to do was provide a "dialog template" describing my customization. I could specify the "dialog template" as a resource linked into my application or I could provide an "in memory object" in the `hInstance` field. From some previous experience in C I knew a dialog template was what a Clarion programmer would call a window definition but how to translate that into a linkable resource or memory object was something of a mystery.

A few [Google](#) searches later I found a sample dialog template for modifying a common dialog:

```
IDD_Open_Options DIALOGEX 0, 0, 230, 20
STYLE DS_SHELLFONT | DS_CONTROL | WS_CHILD | WS_CLIPSIBLINGS
FONT 8, "MS Shell Dlg", 400, 0, 0
BEGIN
    PUSHBUTTON "Filter...", IDC_FILTER, 180,0,50,14
END
```

Although the syntax is different it's quite similar to a Clarion window definition. The first line is the window label and the `x` position, `y` position, width and height. The second and third lines are window properties and a font. Finally the fifth line is a button control to go on the window. This dialog template becomes a child of the `OpenFileName` dialog and in-turn is the parent of what ever control(s) you want to add to the `OpenFileName` dialog.

The other important thing about the sample dialog template I found was the word "Dialogex". A few more searches later I found out that to make an in-memory object to describe a dialog template, I had to fill out a `DlgTemplateEx` structure and one `DialogItemTemplateEx` structure for each control I wanted to add. The `DlgTemplateEx` and `DialogItemTemplateEx` structures programmatically describe a `DIALOGEX` type window definition. To test I quickly filled in the structures and pass the information to the `openfilename` structure and called `GetOpenFileName()`. The good news is my computer didn't melt down but the Open File Dialog didn't appear either. I kept getting cryptic errors about an invalid window class. Clearly, a more detailed examination of the structures was needed.

The `DlgTemplateEx` and `DlgItemTemplateEx` structures needed to be in memory allocated by the `GlobalAlloc` API, or in other words in a global memory block. To modify the `OpenFileDialog` the address of the `dlgTemplateEx` structure needs to be passed in the `hinstance` member of the `OpenFileName` group, and the `OFN_EnableTemplateHandle` constant added to the `flags` member of the `OpenFileName` structure as I'll show later.

The first task is to fill out the `DlgTemplateEx` structure for the modification you would like to make to the common dialog. It is pretty straight forward for the purpose at hand:

```
Dlgtemplateex group
dlgver          ushort !must be 1
signature       ushort !must be 0FFFFH
dwhelpid        long    !0 unless you want to link a help topic
dwExStyle       long    !almost always 0
dwStyle         long    !ds_shellfont+DS_Control+WS_Child+WS_ClipSiblings
cDlgItems       ushort !1 since we are only adding 1 control in this example
x               short   !0
y               short   !0
dlgwidth        short   !230 is the width of the ofn dlg but this value is ignored
dlgheight       short   !50 - any value you wish, 50 worked for me
Menu            long    !0 can be either a pointer to a wide string or resource id
WindowClass     long    !0 for default window class
DialogTitle     long    !0 pointer to a wide string for a
                end
```

The `dlgheight` member is the only value that you may want to vary. It sets the height of your addition to the open file name dialog in dialog units.

The last three items as well as many items in the `DlgTemplateEX` structure can be either a resource id or the address of a wide string (Unicode). There are many items in the `dlgtemplateex` and `dlgitemtemplateex` structures that can be represented by either a resource ID or a pointer to a wide string. Since Clarion does not support numeric resource ids as far as I know, if you wanted to include a resource in one of these structures the appropriate way to do it would be to:

1. Create a wide string containing the name of the resource (`MyIcon_ico` for a icon called `MyIcon.ico` that is linked into a Clarion program program). Use my [strcl](#) supplied with many of my COM articles to make the wide string.
2. Put the address of the wide string into the `DlgTemplateEx` or `DlgItemTemplateEX` structure.
3. When done, free the memory used to construct the wide string. Unlike COM is doesn't matter what method is used to allocate memory for the wide strings.

The only other item worth mention is the `dwStyle` line. The constants listed are all required but you may wish to add additional dialog styles. To find a complete list of dialog styles search MSDN for "Dialog Box Template Styles".

The `DialogItemTemplateEX` structure is a bit more complex.

```
dlgItemTemplateEX Group
dwhelpid        long    !0 unless you wish to link a help item
dwExStyle       long    !usually 0 or any of the ws_ex styles in MSDN
dwstyle         long    !WS_Child + anything else you wish
x               short   !x position
y               short   !y position
w               short   !width
h               short   !height
controlid       long    !max of 256 on 95/98/me; Like a clarion FEQ; stay less than
400H
WindowClass     long    !id or wide string pointer
```

```
Title           long !id or wide string pointer - example: button text
extracount      ushort!extra data to be passed to the control, usually zero
end
```

The first item to settle on is what kind of control you want to create. To tell Windows what kind of control you want to create you specify a window class name. Windows has many prebuild window classes you can choose from. Several of the more common class names have predefined constants that make it easy to specify one of them:

Class	Constant	Screen appearance
Button	80H	Push button, check box, radio control
Edit	81H	Entry control
Static	82H	Image, Prompt
List Box	83H	List Box
Scrollbar	84H	Scroll bar
Combo Box	85H	drop list or combo box
Custom	none	pass the address of a wide string in WindowClass

Once you have specified a constant from the table above or the address of a wide string containing the name of a registered windows class in the WindowClass member for the DlgItemTemplateEx structure, to further clarify the screen appearance you want, add appropriate constants to the dwstyle member. The dwStyle member includes any window style you wish to specify ORed together. For example, in the code sample with this article I wanted to add a button. For the windowclass member, I passed 80H from the table above. I then searched in MSDN for a list of appropriate window style constants for buttons by doing a search for "Button Styles". From the list of available button styles the only button style that applied was BS_PushButton. If I wanted a check box instead of a push button I would have picked BS_CheckBox. I then added the BS_PushButton style to the dwStyle member of DlgItemTemplateEX. For virtually all controls I also add the WS_Child + WS_TabStop + WS_Visible styles. The WS_Child style is required and the other two add the control to the tab order and make it visible – generally good things. The dwEXStyle is virtually always zero unless you have a special purpose. See the example below for a summary of field values used to create the button.

The usage of the Title field is control type specific. For a control that can have text (for example button or static), convert the text to a wide string and pass a pointer to the wide string text in the title member.

The controlid field is very much like a field equate label in clarion. I generally pick a number less than 256 as on some operating systems 255 is the highest number that can be used. For my code sample with this article I picked a control id value of 1. Also beware that in MSDN and other places the documentation for DlgItemTemplateEX may incorrectly show the controlid field as a word. It must be 4 bytes for the code to work. Remember when I said earlier I was getting errors about an invalid window class (window class is the field just after controlid) when the window class clearly was valid? The error went away when I defined the control Id as a long rather than a word. Microsoft shows the correct size for the field in article Q140725. I only wish I could find Bill Gates' personal email address so I could send him a personal thank-you for that documentation error that cost me so much time and frustration.

The common dialogs use control ids in the range 400 – 4FFH as shown in Dlg.h for the predefined controls. You can find the control ids of any of the items on the standard dialog windows by searching the same folders where .h files are for *.dlg. The fileopen.dlg file has the control ids for the fileopen dialog and dlg.h has the numeric value for the control ids.

Turning to the sample code with this article in the dlgtplcl.clw file there is a method called Test. It contains the code needed to display an Open file dialog with a button on it called Preview. When preview is pushed the currently selected file

is passed to an EXE that displays the TIFF and allows paging through it.

With in the test method, the call to the .Init method passes all the information to set up the dlgTemplateEX structure:

```

if SELF.Init(1,| !debug=true
  DS_ShellFont+DS_3DLOOK+DS_Control+|
  WS_child+Ws_ClipSiblings,|
  0,| !windows exStyle
  230,| !width
  20,| !Height
  0,| !x
  0,| !y
  0,| !helpid - not used
  0,,| !menu resourceid=0 & menuresourcenamename (omitted)
  0,,| !windowclassid =0 and windowclassname (omitted)
  ,| !dlgtitle
) then Message('init failed')

```

The following line calls the AddItem method to add a push button to the dialog template. AddItem can be called as many times as you wish to add as many controls as you wish to the dialog template.

```

if SELF.addItem (1,| !control id
  0,|!help id
  WS_Child+BS_PushButton+ws_visible+ws_tabstop,|!window style
  0, ! ws_ex style
  60,0,50,14,|! x,y,w,h
  80H,,|! window class id, omit window class string
  0,'&Preview',|!omit resource id, provide string for title/text
  0,0| !extra bytes and pointer to extra data
) then message('Item addition failed')

```

The code in both init and additem stores the data passed and counts how much memory will need to be allocated to build the DialogTemplateEx and DialogItemTemplateEX structures but does not actually build the structures.

When on the next line after a successful call to init and additem, GetHandle() is called. GetHandle allocates memory with GlobalAlloc and builds the memory structures using the data stored by init and additem.

These memory structures have some unusual memory alignment requirements. In a number of places the next byte written needs to be aligned on a dword or 4 byte boundary. This means the memory address must be evenly divisible by 4. The code to do the required memory alignment is in the Write() method.

Once the GetHandle() method of dlgTp1Cl is successfully called you have the handle to a global memory object containing the dialog template for the Preview button. The code that uses it is in the Test method:

```

Of Event:Selected
  ofn.lStructSize = size(OPENFILENAMEType)
  if vernum < oswin2k then
    ofn.lStructSize-=12
  end
  ofn.hinstance = globallock(hglobal)
  ofn.hwndOwner = 0{prop:handle}
  ofn.lpstrFile = address(szFile)
  ofn.nMaxFile = size(szFile)
  ofn.lpstrFilter = address(StrFilter)
  ofn.nFilterIndex = 1

```

```

ofn.lpstrFileName = 0
ofn.nMaxFileName = 0
ofn.lpstrInitialDir = address(cinitial)
ofn.lpfnHook =address(explorerhook)
ofn.Flags = OFN_PATHMUSTEXIST + OFN_FILEMUSTEXIST |
            + OFN_HIDEREADONLY +OFN_ENABLETEMPLATEHANDLE |
            + OFN_ENABLEHOOK + ofn_explorer
if GetOpenFileName(address(ofn)) then
    message('file selected, no error')
else
    message('Error='&CommDlgExtendedError())
end
globalunlock(hglobal)

```

GlobalLock take a global memory handle as a parameter and converts it to a memory address. OFN is a OpenFileName structure. As soon as the open file dialog closes globalunlock is called to unlock the memory. By unlocking the memory windows is free to relocate your data to help compact memory. When you are done with the Open file dialog, free the global memory handle by calling the dlgTplCl.kill() method:

```

of event:closewindow
    SELF.Kill() !frees the global memory block

```

The one last piece of code in DlgTplCl.Clw is the ExplorerHook procedure. The goal of that piece of code is to respond to button clicks. After all, creating a button is not much use unless you do something in response to the button click. When a button is pushed windows generates a wm_command message. When I receive a WM_command message I check to see if it is from control id 1, my preview button:

```

of wm_command
if band(pWparam,0FFFFH)=1 then
    !1=controlid for the button - button pushed

```

If so I need the name of the file currently selected. The dialog template I have created is a child of the Open file name dialog. So to get the file name from the parent open file dialog I send it a CDM_GetFilePath message. My hook procedure gets the handle for my dialog phdlg passed to it. I use the windows API getparent() call to get the handle to the parent OpenFileName dialog. That message is defined in the OpenFileName documentation in MSDN. The code looks like this:

```

if sendmessage(getparent(phdlg),cdm_getfilepath,|
    size(cFilename), address(cfilename)<>0 then
    if instring('.TIF',upper(cfilename),1,1)<>0 then
        run(cpgmpath & 'gp.exe ' & cfilename,1)
    end
end
end

```

Figure 1 shows the modified dialog. Note the Preview button in the lower left hand corner.

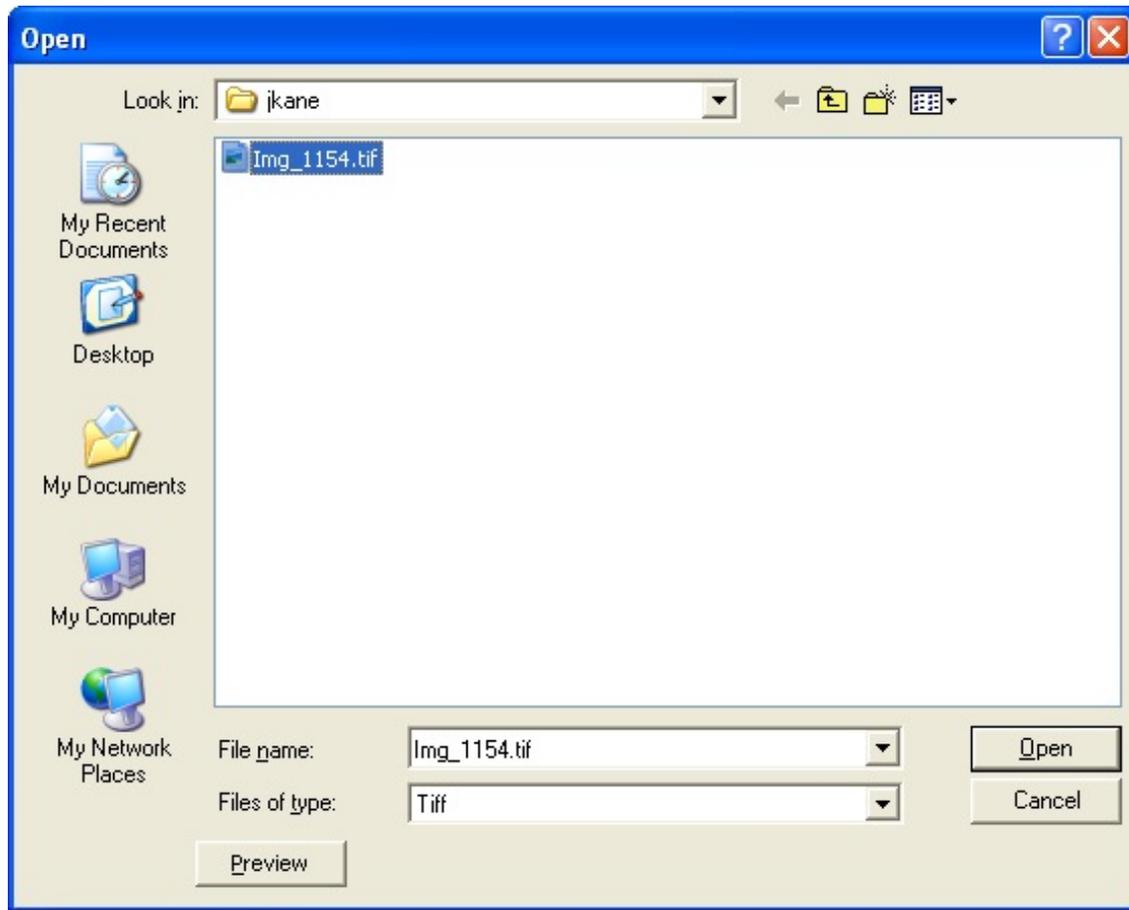


Figure 1. The file dialog with added Preview button

If you select a file and click the Preview button, the file name is passed as a parameter to a program called GP.EXE; the code then waits until GP.EXE is done. While GP.EXE is running, the file open dialog is locked and the screen will not repaint.

The program GP is a simple TIFF file viewer written with GDIPlus, in Clarion. It requires GDIPlus.dll. Because it relies on GDIPlus it only runs on Windows 98 and higher. I have not explored which TIFF format variations GDIPlus can handle and which it cannot. Thus if you find a TIFF that cannot be viewed I would not be too surprised. Also if the TIFF file is corrupt or not truly a TIFF file despite the .TIF extension, a error message, typically out of memory, is displayed. I do not recommend using GP.exe in a commercial project due to its simplistic nature. Perhaps someone will be inspired and write a better TIFF viewer for us. After all, how hard can it be?

[Download the source and DLLs](#)

[Download the source only](#)

[Jim Kane](#) was not born any where near a log cabin. In fact he was born in New York City. After attending college at New York University, he went on to dental school at Harvard University. Troubled by vast numbers of unpaid bills, he accepted a U.S. Air Force Scholarship for dental school, and after graduating served in the US Air Force. He is now retired from the Air Force and writing software for [ProDoc Inc.](#), developer of legal document automation systems. In his spare time, he runs a computer consulting service, Productive Software Solutions. He is married to the former Jane Callahan of Cando, North Dakota. Jim and Jane have two children, Thomas and Amy.

Reader Comments

[Add a comment](#)

Clarion Magazine

A Better DosFileLookup

by David Harms

Published 2005-05-13

In the Clarion application with which I maintain the Clarion Magazine database I use the `ABC DosFileLookup` template to add the full pathnames of articles, PDFs, Zips, and other files to the articles table. That table is also read by a web application on a Linux server (in fact, the database is also on the Linux server); using `DosFileLookup` saves me having to type in each filepath, possibly introducing errors. But `DosFileLookup` doesn't do everything I want, partly because I actually only want to use part of the path, and partly because Linux uses a forward slash to separate directories, and Windows uses a backslash. As well, there are a few other features which the `DosFileLookup` does not support, and which I would find useful.

Whenever some standard Clarion template doesn't do everything I want it to do, my first response is to write some embed code to make up for the deficiency. That's fine as long as I only need that functionality in one place, but as soon as I need it in two or more places, it's time to make some modifications to the class or template, so the new or modified code only exists in one location. I investigated what `DosFileLookup` really does, and I discovered that it's simply a wrapper for an `ABC` class called `SelectFileClass`, which does all the actual work of looking up DOS files. The template makes that class easier to use within the AppGen.

As it turns out, the way to achieve what I wanted was to enhance `SelectFileClass` by *deriving* a class from it, and telling the template to use that class instead of `SelectFileClass`. Before I show you how to derive the class, let me explain how I arrived at that conclusion.

Derived class or modified template?

In the Legacy (Clarion) template chain, the templates generate massive amounts of code – in fact, all code in a Legacy app is either template-generated, or source code you've written (or otherwise obtained). In `ABC`, the templates generate much less code, and are on the whole more of a friendly layer on top of the `ABC` class library. It's the class library that does the real work, not the template-generated code, which mainly consists of creating instances of objects, arranging them in some useful combinations, and calling methods on those objects.

There is actually no good way to make incremental changes to the `ABC` templates. Your choices are to modify the existing templates, or to create similar templates with the additional functionality. In the first instance, you will have problems when upgrading Clarion (either a patch will fail to run, or you will lose your changes on a major upgrade); in the second, you will end up duplicating a lot of code.

There is, however, a relatively easy way to enhance the classes and have those enhancements available to any application. You simply create your own class which derives all of the functionality of an existing class, and which adds the desired behavior. You can then easily tell your application (that is, the `ABC` templates) to use your class instead of the standard `ABC` class, either on a case-by-case basis, or the whole application (depending on the kind of class you're dealing with).

Since `SelectFileClass` is associated with a control template, I'll be specifying my own derived class in each instance of that template. That's a very small amount of work, compared with having to repeat (and maintain) embedded code all over the place!

cciSelectFileClass requirements

My derived class, which I'll call `cciSelectFileClass`, must have the following new functionality:

- retrieve the size of the file
- return a relative path instead of an absolute path (the directory structure for the Clarion Magazine web site on my Windows development machine matches the directory structure on the ClarionMag Linux server, sort of: the filepath `x:\cmag\v7\v7n05selectfile.html` corresponds to a filepath of `/cmag/v7/v7n05selectfile.html` on the server, which gets translated to `http://www.clarionmag.com/cmag/v7/v7n95selectfile.html`).
- translate Windows style forward slashes into Unix-style backslashes (see above).
- remember the last folder from which I selected a file, so I don't have to change folders so much

All ABC-compliant classes follow the same general approach: the class is declared in an INC file, and the method source goes in a CLW file, and these are usually kept in the `libsrc` directory. My derived `SelectFileClass`, which I call `cciSelectFileClass`, is declared in `cciUtil.inc` as follows:

```
!ABCIncludeFile

  OMIT('_EndOfInclude_',_CCIUtilitiesPresent_)
_CCIUtilitiesPresent_ EQUATE(1)

  INCLUDE('ABUTIL.INC'),ONCE

cciSelectFileClass      class(SelectFileClass),TYPE,|
                        MODULE('CCIUtil.CLW'),|

LINK('cciUTIL.CLW',_ABCLinkMode_),|
                                DLL(_ABCDllMode_)

useUnixBackslash        byte(false),private
IniFileName              string('./SelectFile.INI {30}'),private
fileOpenError           cstring(200)
baseFolder               cstring(File:MaxFilePath+1),private
rememberLastFolder      byte(true),private
fileSize                 long,private
ask                     procedure(byte keepDir=0),STRING !overrides parent method
getFileSize              procedure,long
setBaseFolder            procedure(string folder)
setIniFileName           procedure(string filename)
setRememberLastFolder   procedure(byte remember)
setUseUnixBackslash     procedure(byte backslash)
                        END

_EndOfInclude_
```

The first three lines of text, and the last line, are modeled on every other ABC include file. The `!ABCIncludeFile` statement tells the AppGen that this file is ABC compliant, and the `OMIT`, the following label, and the `_EndOfInclude_` statement are compiler tricks to avoid the file being included more than once, which would cause duplicate symbol errors.

The fourth line of text is an `INCLUDE('ABUTIL.INC')` statement. `ABUTIL.INC` is where the original

SelectFileClass is declared.

The class declaration is worth a close look. `cciSelectFileClass` is derived from `SelectFileClass`, has the `TYPE` attribute (when this class is used by the template, yet another derived instance will actually be allocated memory), a `MODULE` statement specifying where to find the method source, and a `LINK` statement, which tells the compiler it needs to link in the method source. Finally there is a `DLL` attribute with the `_ABCDllMode_` parameter, necessary for proper functioning in a multi-DLL environment.

The source file containing the methods is `cciutil.clw`, and it is also quite straightforward, and a little simpler. It begins with an empty `MEMBER` statement, which means the source module can be used with any application. The apparently empty `MAP` is not in fact empty, as several standard Clarion source files are automatically included in any `MAP`, such as `BUILTINS.CLW`.

```
MEMBER

MAP
END

INCLUDE( 'CCIUTIL.INC' )
```

I need a variable and a file structure so I can open the selected file to determine its size. I could also do this in fewer lines of code with the Windows API, but this all-Clarion approach is still pretty easy and quick.

```
DosFileName  STRING(File:MaxFilePath),STATIC

DosFile      FILE,DRIVER('DOS'),PRE(DOS),NAME(DosFileName)
Record       RECORD
Fl           STRING(1000)
            END
            END
```

The following are some simple "getter" and "setter" methods. I could also avoid the methods and simply use properties, but I generally set all properties `PRIVATE` or `PROTECTED` and use methods. Although there's no critical reason to do so in this class, I find it's a good habit to adopt. There will come a time when you want to exercise specific control over what data is placed in a variable, and a setter method lets you do that.

```
cciSelectFileClass.SetRememberLastFolder PROCEDURE(BYTE save)

CODE
self.rememberLastFolder = save

cciSelectFileClass.SetBaseFolder          PROCEDURE(STRING folder)

CODE
self.baseFolder = clip(folder)

cciSelectFileClass.getFileSize            PROCEDURE !,LONG

CODE
return self.FileSize

cciSelectFileClass.setUseUnixBackslash    PROCEDURE(BYTE backslash)
```

```
CODE
self.useUnixBackslash = backslash
```

All of the real work in this class is done by the Ask method. If you look at the original `SelectFileClass` definition, you'll see that this Ask method is declared in identical fashion. That means that `cciSelectFileClass.Ask` *overrides* `SelectFileClass.Ask`, and will be called in its place. That's exactly what I want to have happen, since the template calls the Ask method.

```
cciSelectFileClass.Ask PROCEDURE(BYTE KeepDir=0)

Result    CSTRING(File:MaxFilePath)
x         long
CODE
```

There are only two variables declared, one for the full name and path of the selected file, and the other a counter variable. Although I occasionally use implicits for something as simple as an x integer, declaring the variable is still always the safest route.

The first block of code preloads the default directory from the app's INI file, provided you've instructed the class to remember the previous directory from which you selected a file:

```
if self.rememberLastFolder
    self.defaultDirectory = GetINI('cciSelectFileClass','baseFolder'|
        ,,self.INIFileName)
end
```

Since `cciSelectFileClass.Ask` overrides the parent Ask method, you need to call the parent explicitly.

```
Result = PARENT.Ask(KeepDir)
```

Once again, if remembering the last folder used, save that value to the INI file.

```
if self.rememberLastFolder
    loop x = File:MaxFilePath to 1 by -1
        if result[x] = '\'
            PUTINI('cciSelectFileClass','baseFolder',sub(result,1,x)|
                ,,self.INIFileName)
            break
        end
    end
end
```

Now get the size of the file selected and save it in the `FileSize` property. If no file was selected or there was some other problem, the file size is set to -1.

```
! Get the file's size
DosFileName = result
open(DosFile)
if errorcode()
    self.FileSize = -1
    self.FileOpenError = Error()
else
    self.FileSize = BYTES(DosFile)
    close(DosFile)
```

```
end
```

Here's where the base folder is trimmed off. This requirement is specific to the way I use the class, i.e. to pick a file on a Windows machine and have the filepath correspond to a URL on a Linux box.

```
if self.BaseFolder <> '' |
    and len(result) > len(self.BaseFolder) |
    and upper(self.baseFolder) = upper(result[1 : len(self.baseFolder)])
    result = result[len(self.BaseFolder)+1 : len(result)]
end
```

Finally, if necessary, change all Windows style backslashes to Linux style forward slashes:

```
if self.useUnixBackslash
    loop x = 1 to len(result)
        if result[x] = '\'
            result[x] = '/'
        end
    end
end
end

return result
```

Using cciSelectFileClass

To use this class, simply copy the two source files (cciUtil.clw and cciUtil.inc) to your Clarion libsrc directory. In a Clarion application, populate the DosFileLookup template on a window, right click on the button created by the template, and choose Actions. Navigate to the Classes tab, as shown in Figure 1.

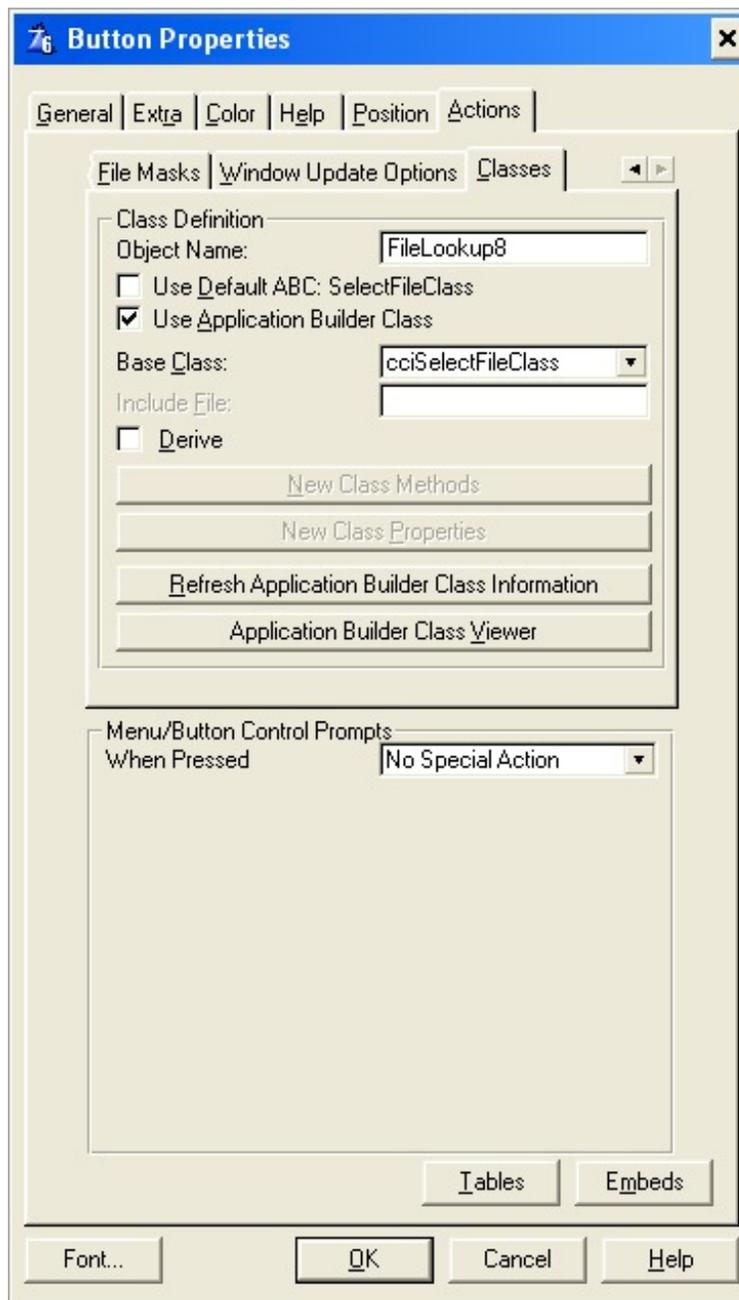


Figure 1. DosFileLookup template actions

Notice that in Figure 1 I have *unchecked* the Use Default ABC: SelectFileClass, and I have chosen `cciSelectFileClass` as the Base Class. Figure 2 shows the class selection drop-down list.

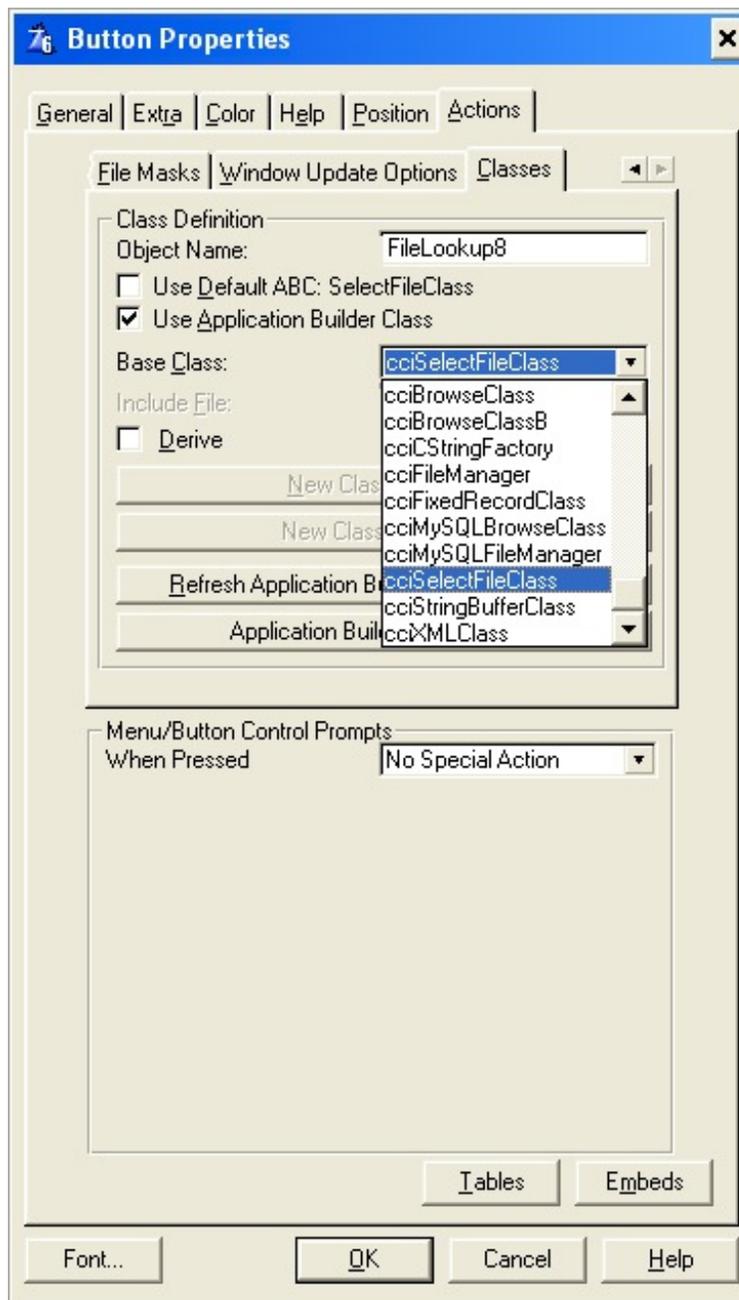


Figure 2. The class selection drop-down list.

If `cciSelectFileClass` doesn't show up in the class list, you may need to press the Refresh Application Class Builder Information button, which will tell the AppGen to re-read all the header files in the `libsrc` directory, and build its list of classes from those containing the `!ABCIncludeFile` comment.

In order to use the new features you will need to enable them. Ideally, these features should appear on the template prompt, but as I indicated earlier, modifying the standard templates has its own problems. A better solution is to create an accompanying extension template, and I'll discuss that in a future article. For purposes of this article, your options are to modify the class to enable certain features by default, or to embed just a few lines of code.

Remember that `cciSelectFileClass` has the `TYPE` attribute – what the `DosFileLookup` template does is create a new class instance for each instance of the template. So really there are three levels in this class hierarchy: `FileLookup8` (using this example) is derived from `cciSelectFileClass`, which is derived from `SelectFileClass`. To embed some setup code, go to the embed list for the `DosFileLookup` template, and choose an embed point in the `Ask PROCEDURE (BYTE KeepDir=0) , STRING` call, before the parent call. You can then embed code similar to the

following:

```
self.setBaseFolder('c:\')
self.setRememberLastFolder(true)
self.setUseUnixBackslash(true)
```

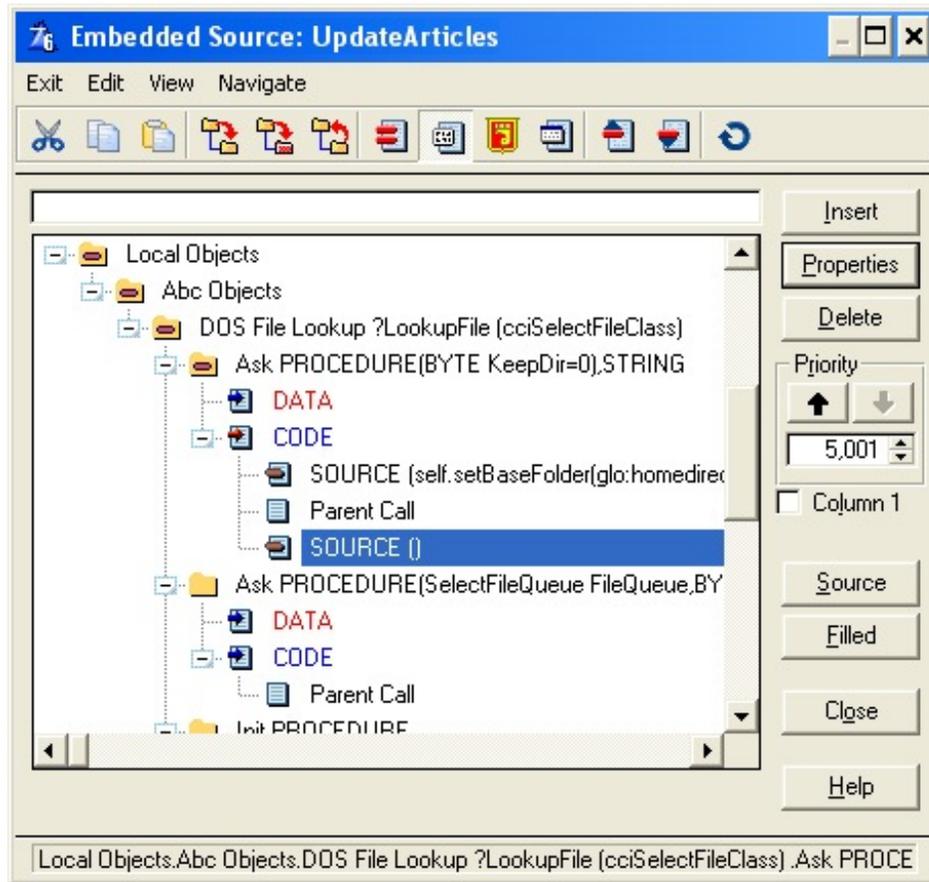


Figure 3. The embed list

From Figure 3 you'll notice there is another `Ask` procedure in the embed list – that method is used when you want to return multiple selections, which is not relevant to the enhancements I've described in this article, exception of remembering the last directory used. I leave the application of that technique to the second `Ask` method as an exercise for the reader.

Summary

One of the great advantages of the ABC templates over the Legacy templates is the ease with which you can make significant enhancements to standard Clarion functionality with minimal effort, and without disrupting the standard Clarion installation. It's quite easy to derive your own ABC-compliant class, have it available within the application generator, and use it in place of the standard ABC class expected by the template. Most importantly, you can use your own derived classes repeatedly without any significant extra work.

[Download the source](#)

[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](#)

- [» Some nice additions to the class that SV should...](#)
- [» The only way I get initialized properties is through the...](#)

Clarion Magazine

Whitemarsh's Use Of Mimer With Clarion

by Michael Gorman

Published 2005-05-19

This is a very brief review of the [Whitemarsh](#) use of the Mimer DBMS. Much more information can be obtained from the [Mimer website](#).

Whitemarsh chose Mimer as the DBMS for use with the demo and default versions of the [Metabase](#) for a number of reasons. These included its very small footprint, ease of installation, ease of use with Clarion, and it's conformance to ISO/ANSI SQL.

The process of obtaining Mimer is simple. Go to the [website](#), download a developer copy, and then start using it. The instructions are quite clear. There is no charge for the developer copy, although it is restricted to 10 concurrent users. It is my understanding that server licenses are about \$2500 for an unlimited quantity of users.

Mimer doesn't come with all the "stuff" that you probably "should never" use anyway, such as a Mimer proprietary query language, Mimer proprietary report writers, and a Mimer proprietary stored procedure language. To compensate for these "missing" pieces, we recommend [WinSQL](#) for the ODBC SQL client, and we use [Crystal Reports](#) for report writing. They more than compensate, and they also provide a level of independence from the DBMS engine.

Clarion & Mimer

Clarion External names are required for all columns. And since we let Mimer do our auto-incrementing on the primary key, we had to the primary key fields Read Only (Column

Properties, Attribute tab) for every table. This causes Clarion to send a null value to Mimer, and Mimer then generates the correct value.

Mimer doesn't return the primary key value for a newly inserted row as does Clarion with TPS. So, to get the value, and to enable the Whitemarsh metabase to be independent of any specific SQL DBMS, we have an autoincrement key for every table, and also a unique key. For example, the Student table has a StudentPkey with StudentId for the autoincrement, and StudentUniqueKey with StudentLast, StudentFirst, StudentMI, StudentBirthDate to frame a unique key for a student. When, in this example you store the student's record, you definitely would provide the StudentLast, StudentFirst, StudentMI, StudentBirthDate. Once you get the okay that the record is stored, you just do a Get with the UniqueKey and then you have the primary key ID value so you can go on and store, for example, the student's address which might be in another table.

The only other thing was that we broke the Mimer SQL script file into three streams, one for the tables, primary keys and unique keys, one for the auto-incrementing commands, and a third file for just the foreign keys. We generated all the SQL scripts with Roberto Artigas's [DCT2SQL](#) script generator. It's quite good.

Mimer has a stored procedure language that is the ANSI standard SQL programming language. We don't use it because none of the other SQL DBMSs have implemented it. Otherwise we would. Mimer has a backup and recovery capability, data export and import, security, and transaction management.

All in all, Mimer is a good SQL DBMS. Once you've implemented it, you can forget about it.

[Michael M. Gorman](#), the founder of [Whitemarsh Information Systems Corp](#), has been involved in database work since 1969. He is a charter member of X3H2, the American National Standards Institute (ANSI) National Committee on Information Technology Standards (NCITS) H2 Technical Committee on Database, and its secretary since 1978.

Reader Comments

[Add a comment](#)

Clarion Magazine

Easier Version Control: Using TortoiseSVN With Subversion

by Mark Geisinger

Published 2005-05-19

Version Control is Very Complex. Not!

Actually, both of those statements are true. More to the point, a good version control system hides the underlying complexity of revision management, allowing you to easily accomplish routine tasks.

I chose the [Subversion](#) server and the [TortoiseSVN](#) client for my use. Subversion is an open source project whose capabilities suit my needs well. It is distributed as both source code and in binary packages for all popular operating systems. As I have all Linux servers, I chose to compile and install Subversion from source.

The client, TortoiseSVN, is a Windows shell extension distributed as a Windows Installer setup. As a shell extension, TortoiseSVN is available from the Windows Explorer, or any Explorer-like product – just right-click on any file or directory to see the TortoiseSVN extensions (see Figure 1). As the everyday work with the version control client is all about directories and files within a project, good shell integration makes TortoiseSVN very easy to use.

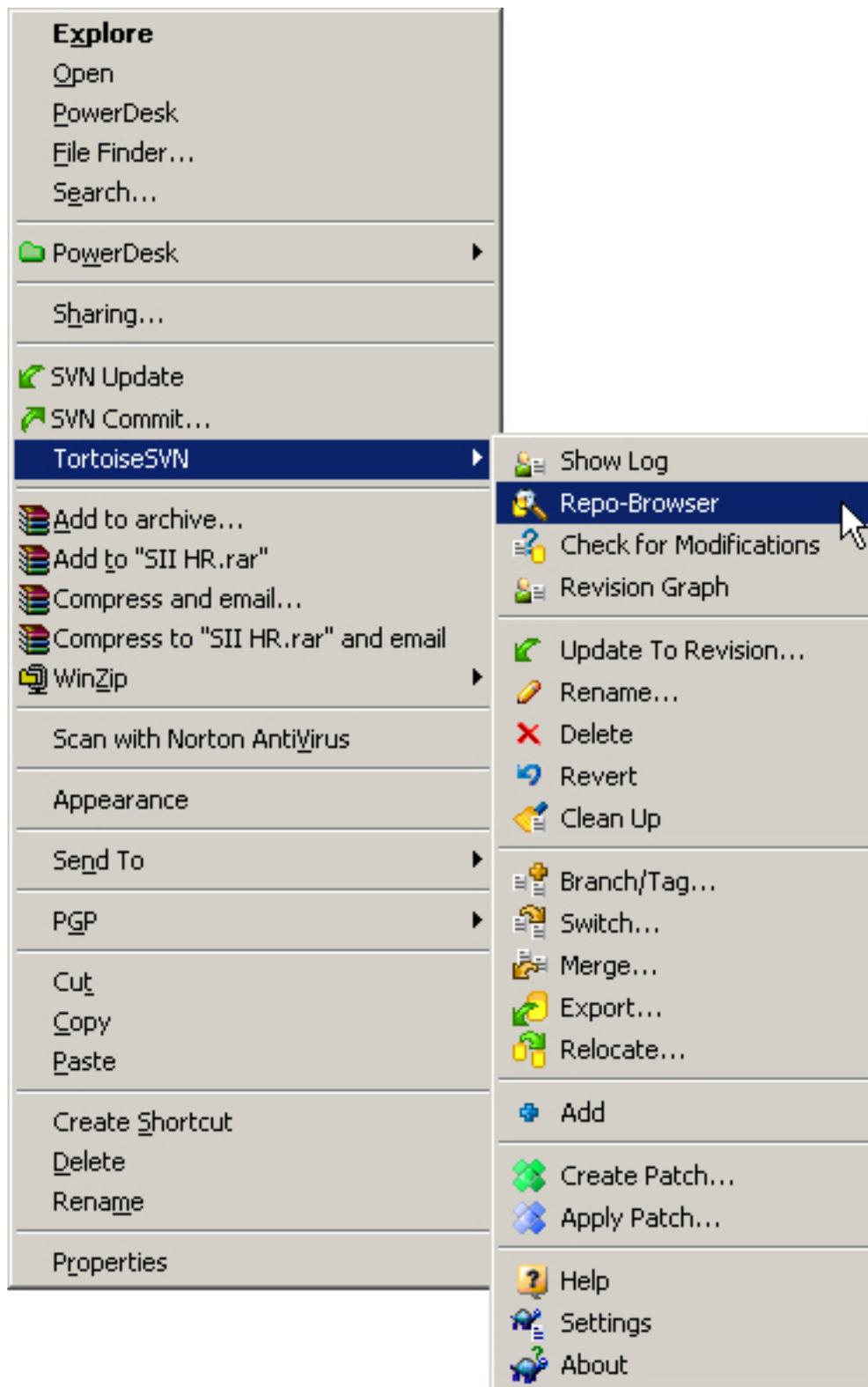


Figure 1. The TortoiseSVN Context Menu

After using TortoiseSVN with Subversion for several months, I have come to the conclusion that version control is *all* about the client. There are many capable servers from which you can choose, but your everyday work is performed exclusively with the client. Select one that both works well and is easy to use, and you may find yourself saving

incremental changes to your work more frequently than you previously performed backups.

Version control is a better backup system

Backup! Backup! Backup! We know it is important. Tape is best; [InBack](#) is awesome. Yet both are an all or nothing method. I have gigabytes of InBack project backups, yet I tend to not use it as frequently as I know I should. At between 60MB and 120+MB each, those project archives tend to be large and time consuming to perform. While I should use it more, InBack has never been a tool I fire up three or four times per day.

As Subversion only stores changes to files, and TortoiseSVN literally makes commits as simple as a mouse click, I have found that these two tools have changed my work habits. While not at all necessary, committing changes several times during a workday has become routine for me. In combination with daily tape backup and mirroring the Subversion repository to a second server, I have both very easy to use version control and a better backup system.

Getting started

There are three prerequisites to be met before proceeding.

- You have installed [Subversion](#) on a server, or on your local machine, or you have access to Subversion running somewhere on a server.
- You have installed the [TortoiseSVN](#) client
- You have created a repository [on your local machine](#), or [on the server](#), or you have access to an existing repository.

These three tasks are easy to accomplish – Windows install programs are available for both Subversion and TortoiseSVN, and creating a repository is clearly explained in the above link.

I have created a new repository on my Subversion server named `cmag.article`. I will begin with the empty repository and go through the process of creating directories, importing a project, and committing revisions to that project. I have chosen a small Clarion project comprised of eight DLLs for use throughout this illustration.

Storing, or *importing*, an existing project in your repository for the first time requires a bit of explanation that will save you some of the trial and error work I did early on. The default behavior of TortoiseSVN when importing a project is to store all files and directories within the directory hierarchy you select. While it is harmless to store unneeded files in your repository, there really is no need to include some files that are created during each build cycle. I recommend you do include all your generated source, though.

Excluding files and folders

TortoiseSVN provides a straightforward set of configuration settings available from the Settings item in its context menu. Exclude / Ignore pattern allows you to specify name patterns used to exclude files and folders from versioning by TortoiseSVN.

Exclude / Ignore pattern applies to the specific Windows User using TortoiseSVN. You can also edit the local Subversion configuration file used by TortoiseSVN, but exclusions set there apply to all users. I find that using Exclude / Ignore pattern works fine for me.

Entering a list of names separated by a space specifies exclusions. DOS file wildcard patterns may be used for files, but do not apply to directories. To exclude a directory, enter its name in the list and all directories of that name are excluded. I found that TortoiseSVN exclusions are case sensitive in the release I started with, so I primarily use a third type of file name specification; the regular expression character set. Some of the file types I exclude are .exe, .dll, .map, .exp, and .lnk.

Example Exclusion Patterns	
[Oo][Bb][Jj]	Excludes the directory obj , OBJ , or oBj
*.\$\$\$	DOS wildcard file name exclusion
*.[Rr][Ss][Cc]	It seems wildcards and regex can be mixed for files

Preparing your project

Backup! Backup! Backup! This is your valuable work and, as you will see in a moment, you will at some point become comfortable enough to completely delete your original

project directory. You don't have to do that, but make an old man happy and backup your work anyway.

I made two mistakes when importing my first project. I assumed that the directory I imported would automatically be used as the working project directory, and I imported a large number of unnecessary files. That includes those mentioned above, along with sample databases, old test code I no longer needed, and the like. Be kind to your server and take this opportunity to clean out some old files you don't need.

Taking a project from non-versioned to versioned requires that you first import it to your repository and then check it out into a *new* directory on your local system. If you are particular about where your working projects reside, you will want a versioned project to exist in the same place your non-versioned one does now. To accomplish this that directory must either exist and be empty, or not exist in which case TortoiseSVN will create it for you. I found that moving a project directory to a safe place, cleaning out unneeded files, importing from this new location, and then checking out to the original project directory worked best for me.

Connecting to and browsing the repository

With your local files and folders ready for import, it's time to have a first look at your repository. Keep the Windows Explorer, or your favorite file manager, handy. Throughout my discussion of TortoiseSVN I will be initiating actions from a file manager.

TortoiseSVN allows you to browse your repository much like you would your local file system. Its simple browse window displays a repository as the familiar directory tree. Don't be disappointed by the minimal set of management functions available in the browser. Everything you need is there.

All of TortoiseSVN's functionality is accessed via the shell's context menu, so it's available from almost anywhere in Windows. When working with specific projects you will most frequently access the shell menu from a project's top level folder. Doing that with an active project folder creates the menu shown in Figure 1, above.

Selecting Repo-Browser from the menu initiates one of two actions. If the current file or folder is part of a project under version control, TortoiseSVN opens the browser with that project's folder selected. If you start the browser from a non-versioned folder, or another location like the desktop, TortoiseSVN presents a window prompting you for an URL to

the repository you wish to browse as shown in Figure 2. TortoiseSVN conveniently maintains an URL history for you. If your repository requires authentication, you will be prompted for your user name and password. TortoiseSVN can optionally save this information for you.

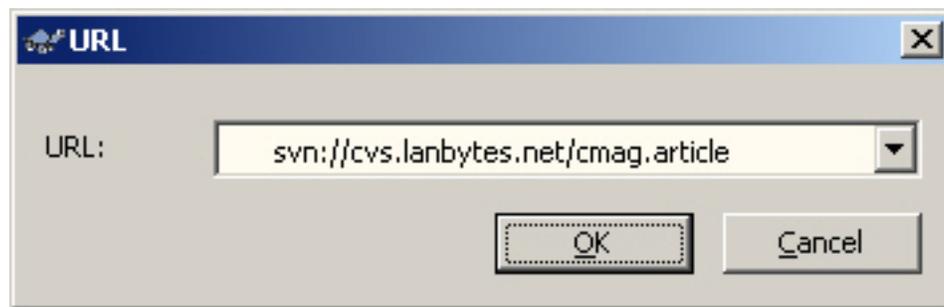


Figure 2. Opening Your Repository with Its SVN URL

Creating your project's folders

Before importing your project, you need to create at least one folder in your repository. Repository actions in the browser are accessed via a context menu, as shown in Figure 3. As in Windows Explorer, actions are taken on the currently selected object.

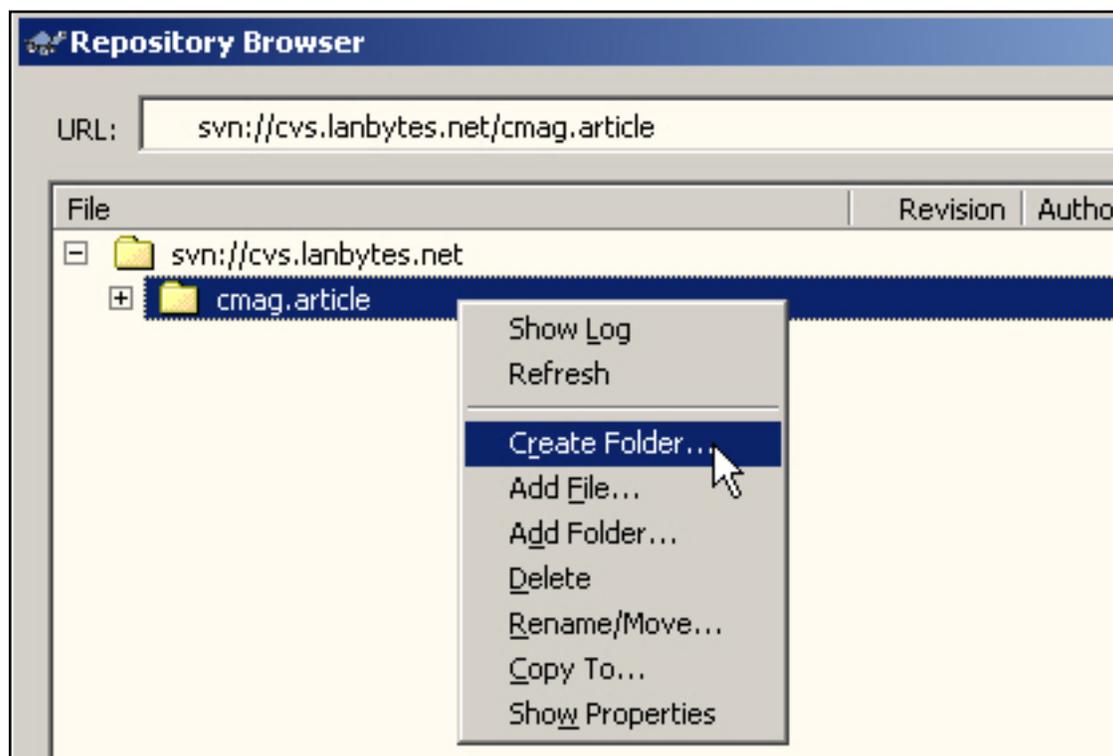


Figure 3. Direct Repository Actions

You are free to create any folder hierarchy with which you are comfortable. Neither Subversion nor TortoiseSVN are sensitive to folder names, or structure. If you haven't worked with a source control system before, I recommend taking some time to consider how the folders will be used. Starting this process with a plan in mind will allow you to make the most efficient use of your repository.

There are some recommended folder layouts, one of which I use here. You will find this explained in some detail in the TortoiseSVN help (Section 4.3.1. Repository Layout) and in the book, "Version Control with Subversion". Both are well written and offer lots of helpful information.

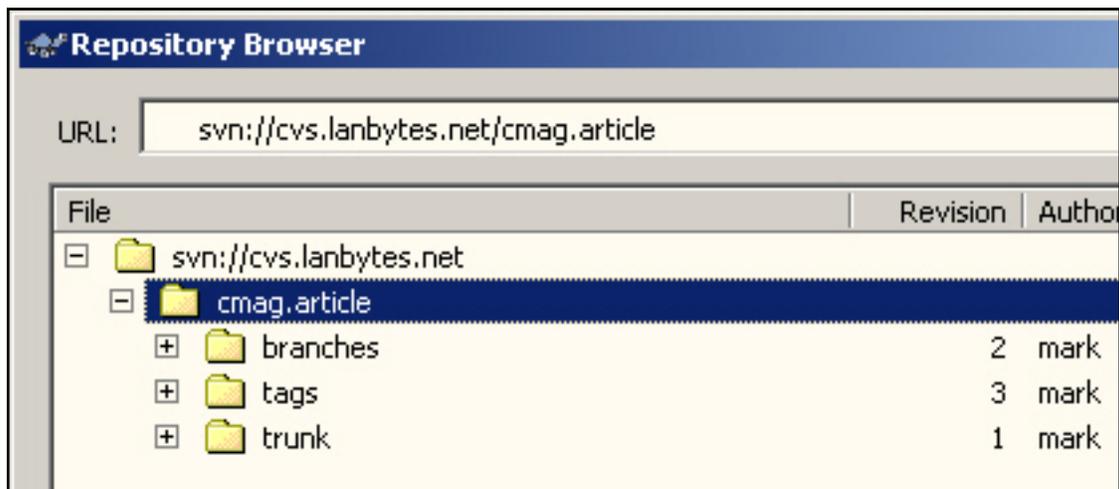


Figure 4. Your Project's Repository Tree

Using TortoiseSVN's Create Folder menu item, I created three top level folders in root of the `cmag.article` repository as follows. Figure 4 shows this directory structure in the repository browser.

Directory	Description
branches	A branch is used to separate changes into a distinct line of development. You might create a branch to test some new development while leaving your existing code base intact.
tags	A tag is a snapshot of a project you wish preserved intact. For example, when you release v1.1.0.0 of a project you might create a tag in this folder named "release.1.1.0.0"
trunk	Contains the current "main line" of development. That is, the revision on which you are currently working.

Importing your project into the repository

Now that you have removed unneeded files from your project and your repository directories are in place, you're ready to import it into the repository database. Figure 5 shows the SII HR project prior to import.

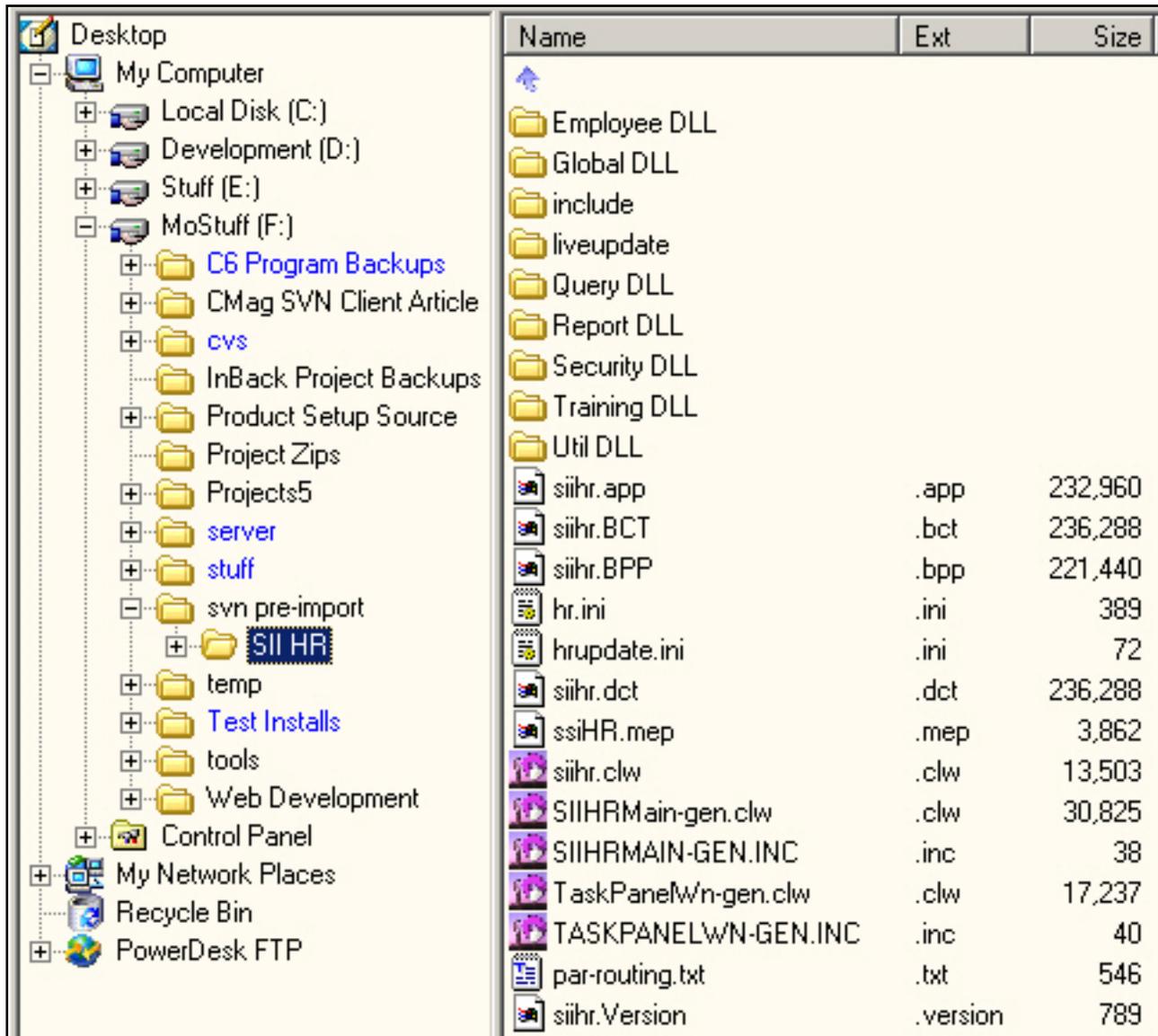


Figure 5. Sample Project Before Import

Tip: TortoiseSVN imports the *contents* of the selected directory recursively into the repository. It does not import the selected directory itself.

To import your project, first select the top level project directory in the Windows Explorer. Right-click to display the context menu, drop the TortoiseSVN submenu, and then select Import. TortoiseSVN opens the dialog shown in Figure 6 where you must provide the full URL to the repository directory into which you wish the project imported.

I am importing to the `trunk` directory, so I selected the saved top level repository URL and typed the directory name to complete the path.

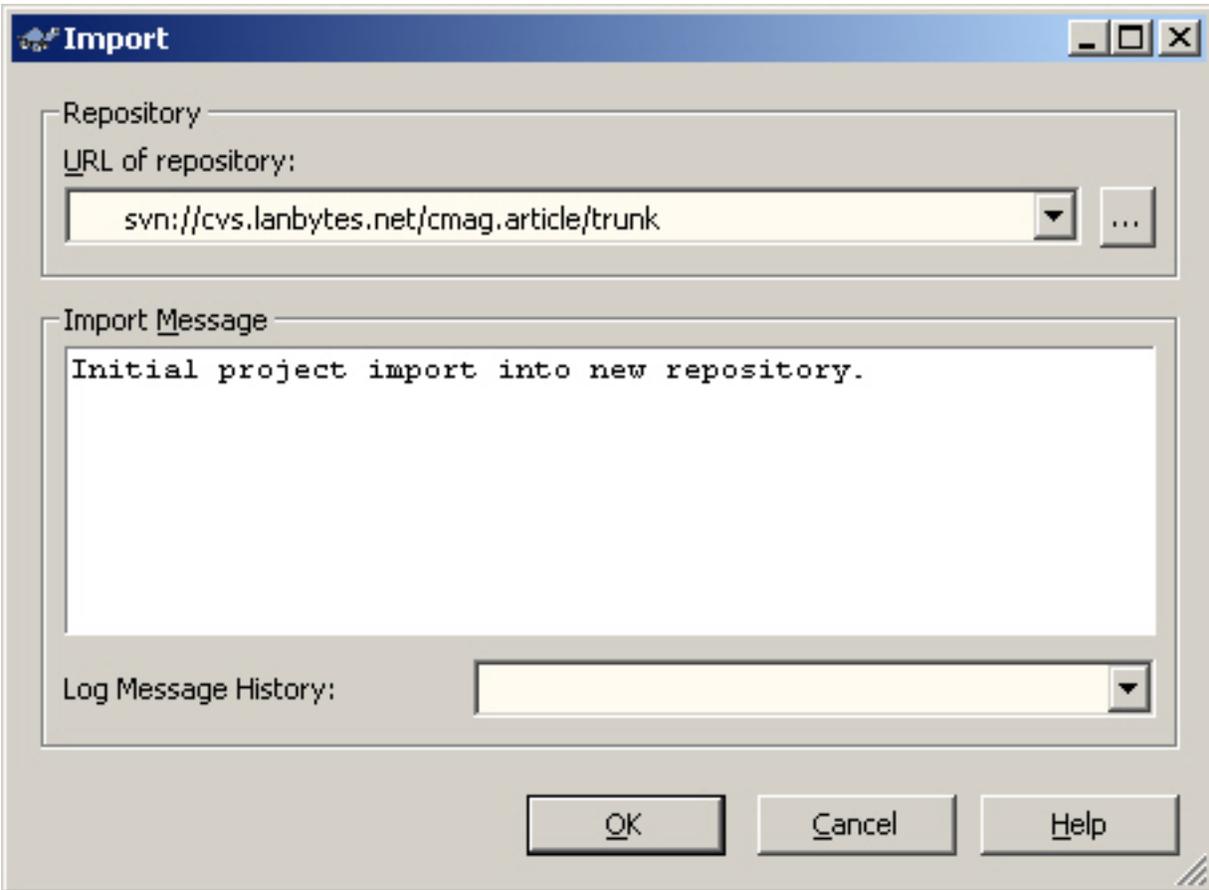


Figure 6. Project Import Into Trunk Directory

Tip: TortoiseSVN seldom requires typing an URL. An alternate method is to select an URL from its history and use the ellipsis button to browse the repository and select a directory.

Time to checkout

With the project safely stored in its repository and local backups available, it's time to checkout the project and return to work. I have chosen to store the working project in a directory named `SII HR`, but in a location other than the original.

Tip: When performing a checkout, TortoiseSVN places all of the project's files into a top level directory of your choice. You can create an empty directory in the Explorer before performing the checkout, navigate to a directory from within the checkout dialog, or enter a path yourself. You

should always checkout into an empty directory, or a new one created during the checkout process.

I performed the checkout by selecting the directory in which I want the SII HR directory created and choosing *SVN Checkout* from its context menu. In TortoiseSVN's checkout dialog, shown in Figure 7, the project trunk directory was selected from the URL history, and the directory name appended to the existing path. As SII HR did not exist, it was created during checkout. Figure 8 shows the SII HR project ready for use.

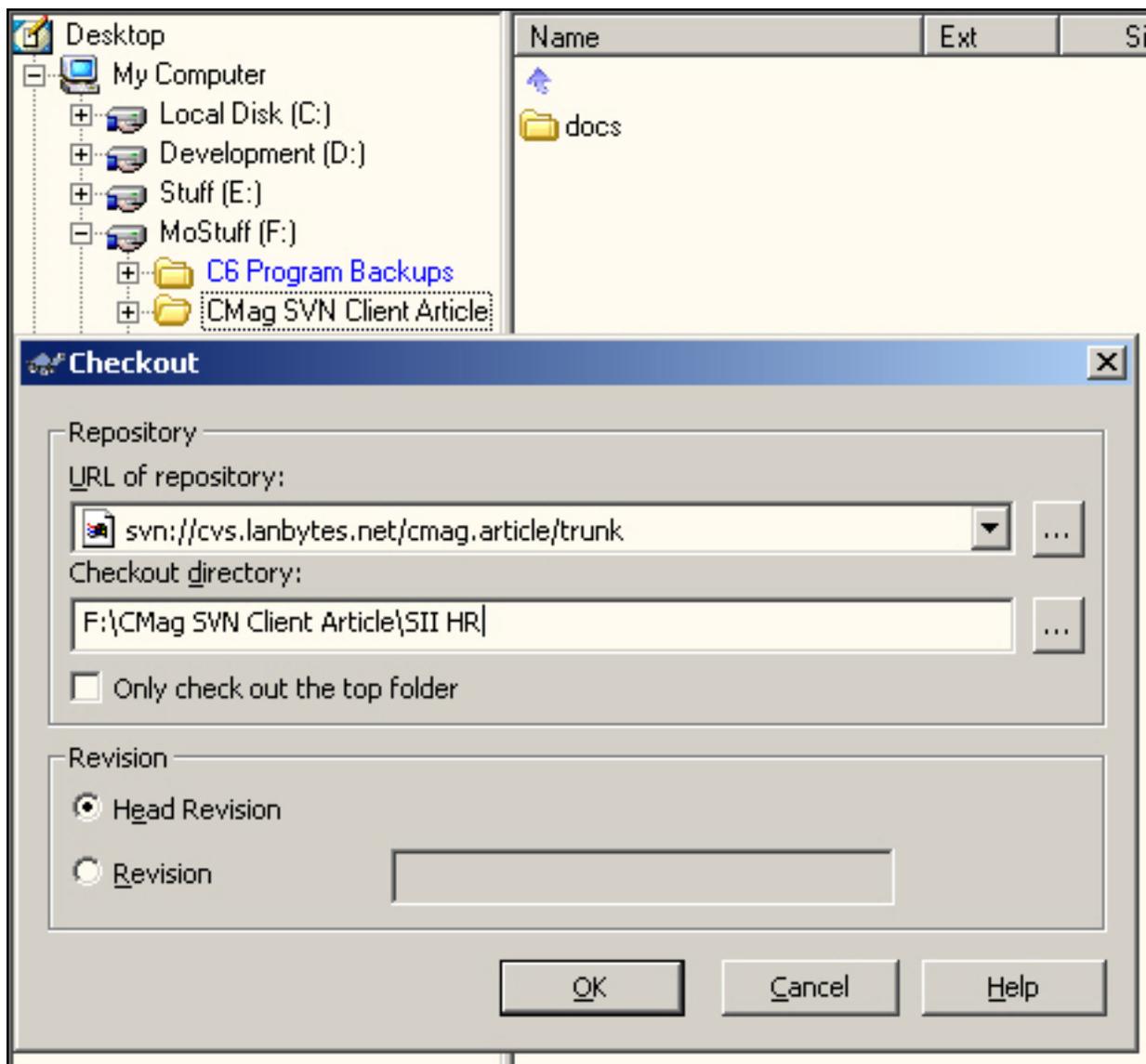


Figure 7. Project Checkout

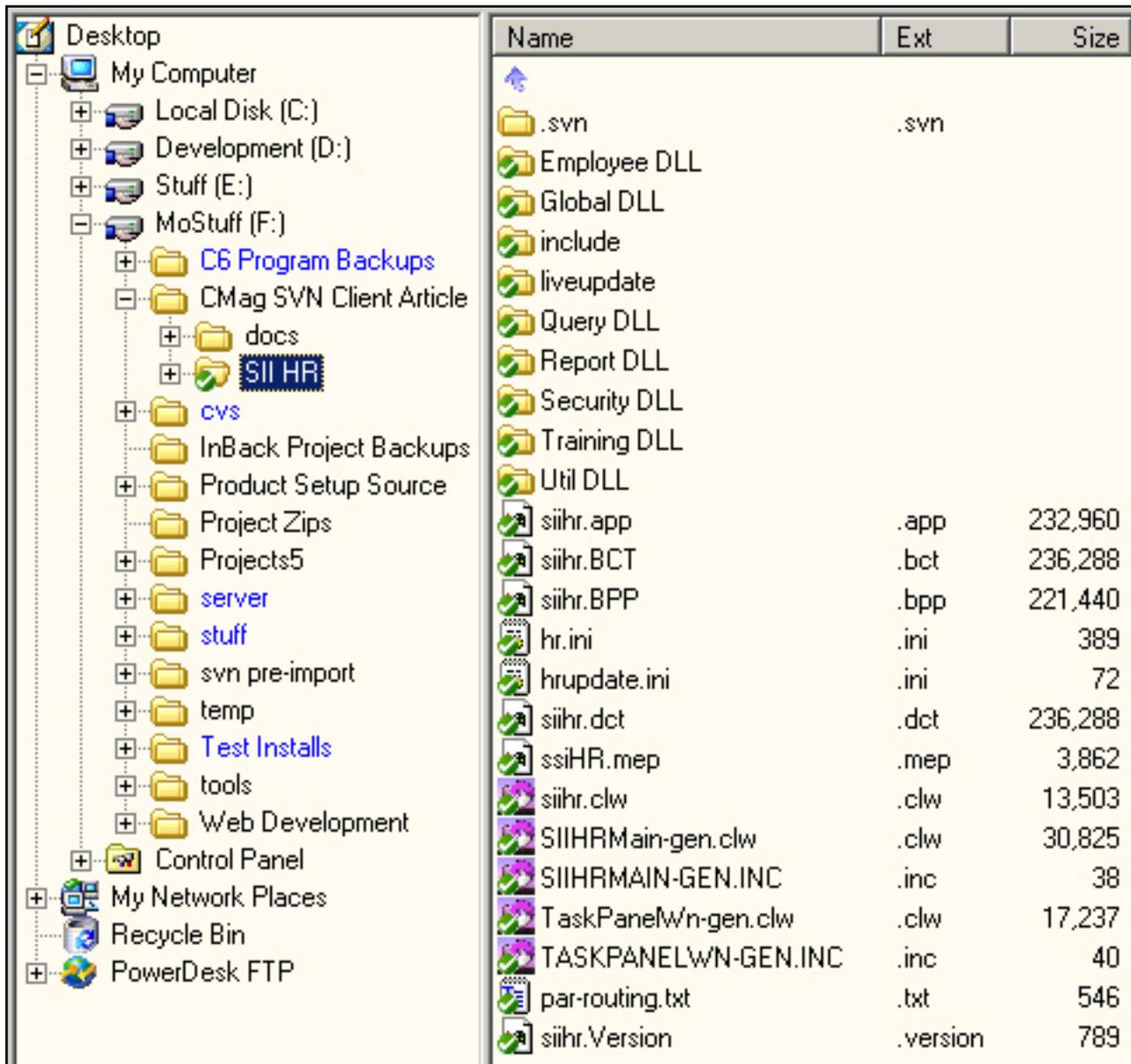


Figure 8. The Project After Checkout

Commit me, please

I've been hard at work on this little project and it's time to save my changes to the repository, or commit the project. Like other actions I have shown you, committing your project requires little more than a couple of mouse clicks. With the project under version control, TortoiseSVN knows its repository location so even the simple step of URL selection is unnecessary.

I need only right-click the top level project directory, SII HR, select SVN Commit, and then enter notes on the work I've done. TortoiseSVN sends only the file changes to the repository where they are efficiently and safely stored.

There is much more you can do with version control and with the tools I have described. I encourage you to shop around and evaluate two, or three, of the most appealing products. For those interested in Subversion and TortoiseSVN, I have included links to their sites and documentation. The TortoiseSVN help is well written and offers information useful to novices and experienced version control users alike. If you're considering setting up your own server, the book [Version Control with Subversion](#) is a must read both for the information specific to Subversion and for the background material it includes.

Resources

- TortoiseSVN <http://tortoisesvn.tigris.org/>
 - TortoiseSVN Documentation <http://tortoisesvn.tigris.org/docs.html>
 - Subversion server <http://subversion.tigris.org/>
 - "Version Control with Subversion" - A Subversion book by Mike Pilato, Brian Fitzpatrick, and Ben Collins-Sussman; treeware available from O'Reilly, plus full sources/HTML/PDF available online
 - <http://svnbook.red-bean.com/>
-

Residing in South Louisiana and the creator of the "How Pork Fat Can Improve Your Life" diet, [Mark Geisinger](#) has been developing software with Clarion since 1993. In addition to software development, he owns a boutique Web and email hosting company.

Reader Comments

[Add a comment](#)

Clarion Magazine

First Look: RADventure's RADProfiler Simplifies Tracing and Debugging

by David Harms

Published 2005-05-20

Many years ago, when the earth was still hot and dinosaurs roamed the land, I began bugging TopSpeed's David Bayliss for the ability to profile my Clarion code, so I could tell how much time any given block of code took to execute. After one particular release (I no longer remember which one – this is, after all, ancient history) David told me to check the libsrc directory for a file called PROFILE.CLW.

I was delighted to find that PROFILE.CLW contained prototypes for two new procedures, `EnterProc` and `LeaveProc`, and an example of how to use these for basic code profiling. All I had to do was set a project pragma, and these two procedures would be called whenever my code entered or left a procedure (respectively). I quickly realized that not only was this technique useful for profiling, but also for creating a tree-view trace of a running application's procedure calls.

My own profiling code, which I presented at DevCon 99, and which I've [written about](#) here in Clarion Magazine, has served me well. But I'm delighted to see that RADventure's free RADProfiler has made great strides forward in adapting the profiling hooks to the task of debugging applications. The key benefits RADProfiler offers, in my opinion, are improved output options (not least is the ability to use [DebugView](#)) and trace filtering.

Installing RADProfiler

RADProfiler is easy to install – simply download and run the zipped installer. The install

program makes the point that you must close all instances of Clarion before proceeding with the installation. The setup information notes also specify that three files will be installed – a template, an INC, and a CLW. You will get another reminder about closing Clarion before the install program registers the RADProfiler template.

Using RADProfiler

Using RADProfiler is almost as easy as installing it. Load your application into Clarion, click on the blue basketball, or choose Application|Global Properties, to bring up the Global Properties dialog. Press Extensions, and add the RADProfiler extension. Figure 1 shows the first tab of the extension's properties.

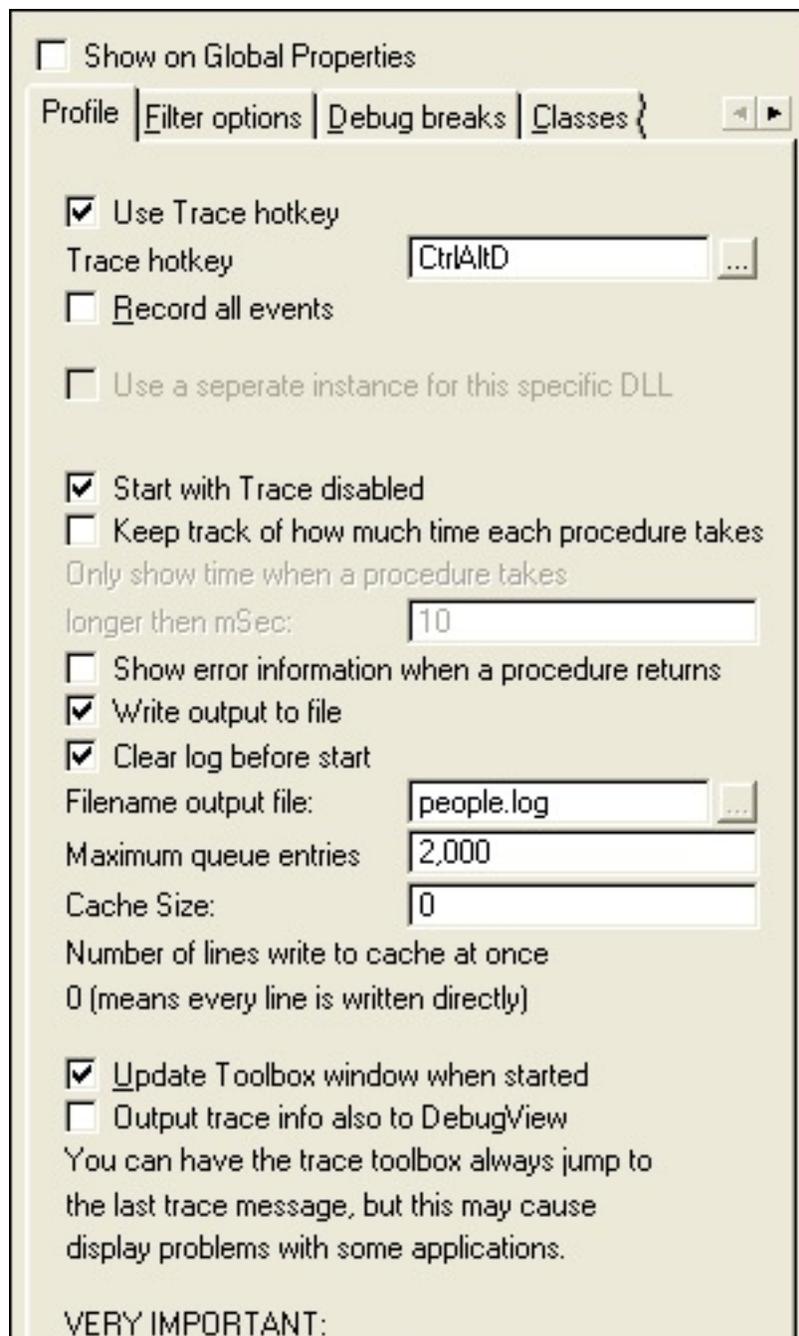




Figure 1. RADProfiler general properties

There are a number of handy options on the global extension. You use the hot key (default is Ctrl-Alt-D) to get access to trace data and settings at runtime. Note that the default setting is to disable tracing at program startup. Any Clarion application, particularly if it's an ABC app, will generate massively long traces. Often it's better to just turn on tracing where and when you want it.

If you press Ctrl-Alt-D while the program is running, you'll see the menu shown in Figure 2. As you can see, some of the runtime options let you override the template defaults.

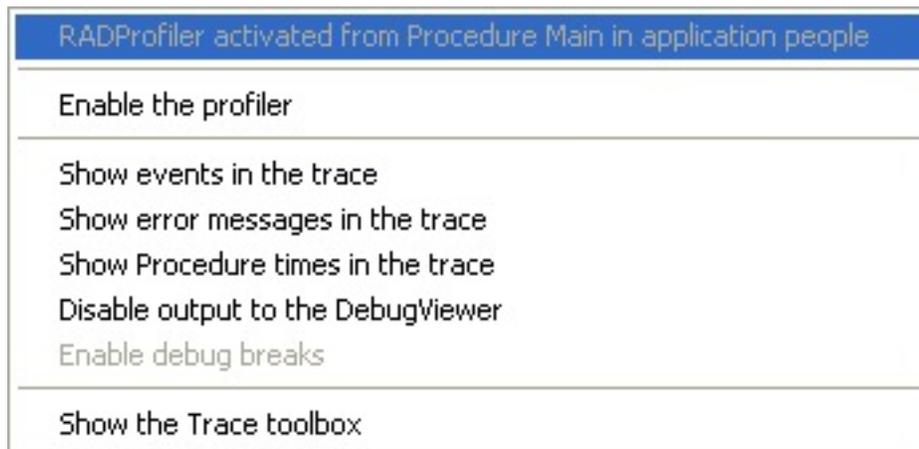


Figure 2. Runtime settings

I won't cover all of RADProfiler's options, as many are self-explanatory, but there are a few I think are particularly important. The cache size matters if you're using RADProfiler to determine the location and cause of a crash via a trace file. If you set the cache to a non-zero value, then you may not get an accurate idea of the last procedure call before the crash.

The option to output the trace log to DebugView is, in my opinion, one of the best reasons to use RADProfiler. When I wrote my own profiling code I used the technique of storing trace calls in a queue, and displaying this queue as needed. RADProfiler does that as well,

but it's often better to view the trace output live, in another application entirely. Enter [DebugView](#), a free utility from [SysInternals.com](#) which intercepts and displays all calls to the Windows OutputDebugString API call.

Tracing ABC calls

Let's face it – following the path of execution in any object-oriented application, with derived classes and virtual methods, can be maddening. DebugView is part of the solution, as you can watch the trace as it happens, and see the results of your interaction with the program. The other part is RADProfiler's filtering capability. Let's say you're having a problem with Edit-In-Place, and you want to know where and when code is executing. Go to the RADProfiler global extension, Filters tab, and enter a filter like the one shown in Figure 3.

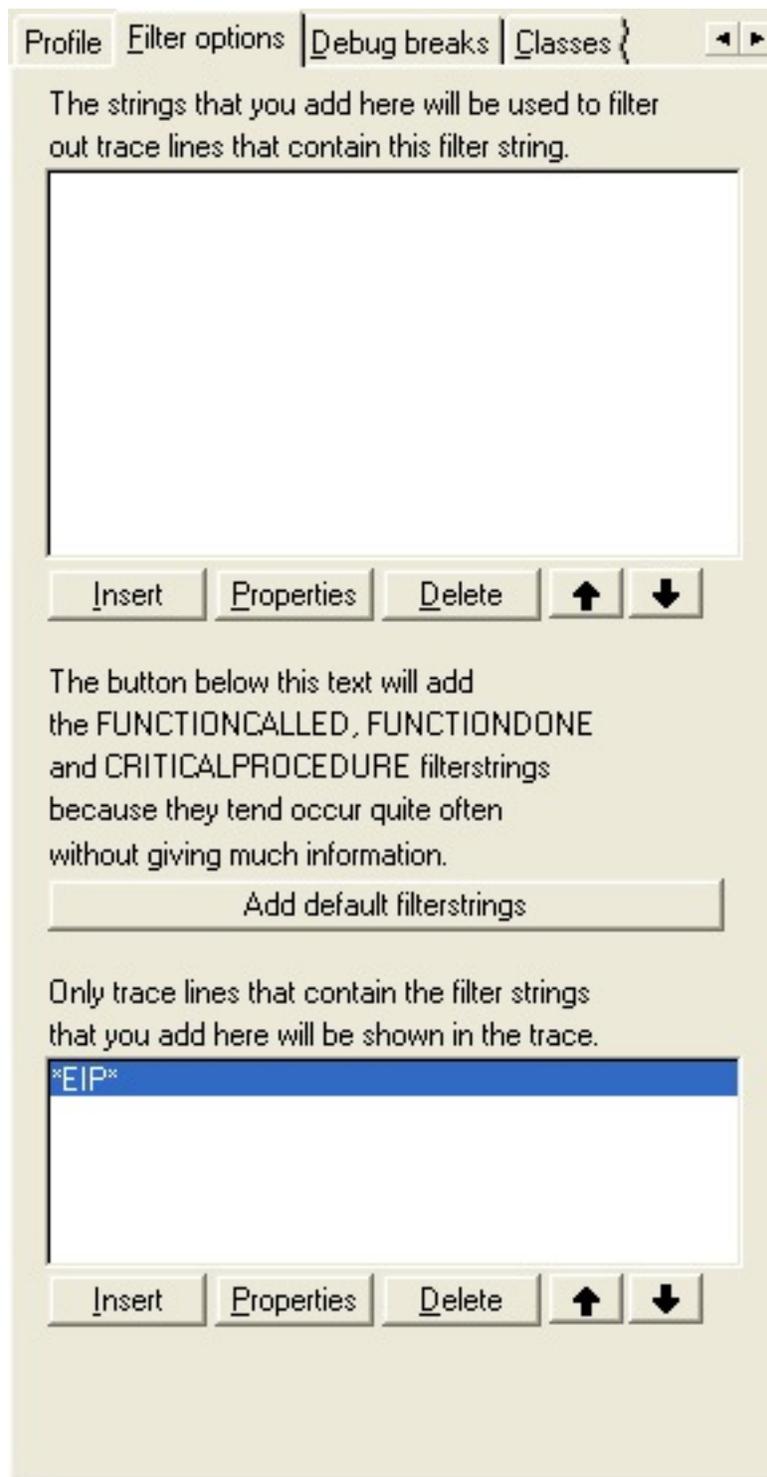


Figure 3. Specifying a filter

Since the class used by ABC is EIPManager, the string *EIP* will match the class name, as well as the ABEIP.INC and CLW files. Launch DebugView, run the application with RADProfiler enabled, and proceed to use EIP. If you have enough screen real estate (this is where multiple monitors come in really handy) you can have your app and DebugView visible at the same time, and you can see which procedures are being called, as shown in Figure 4.

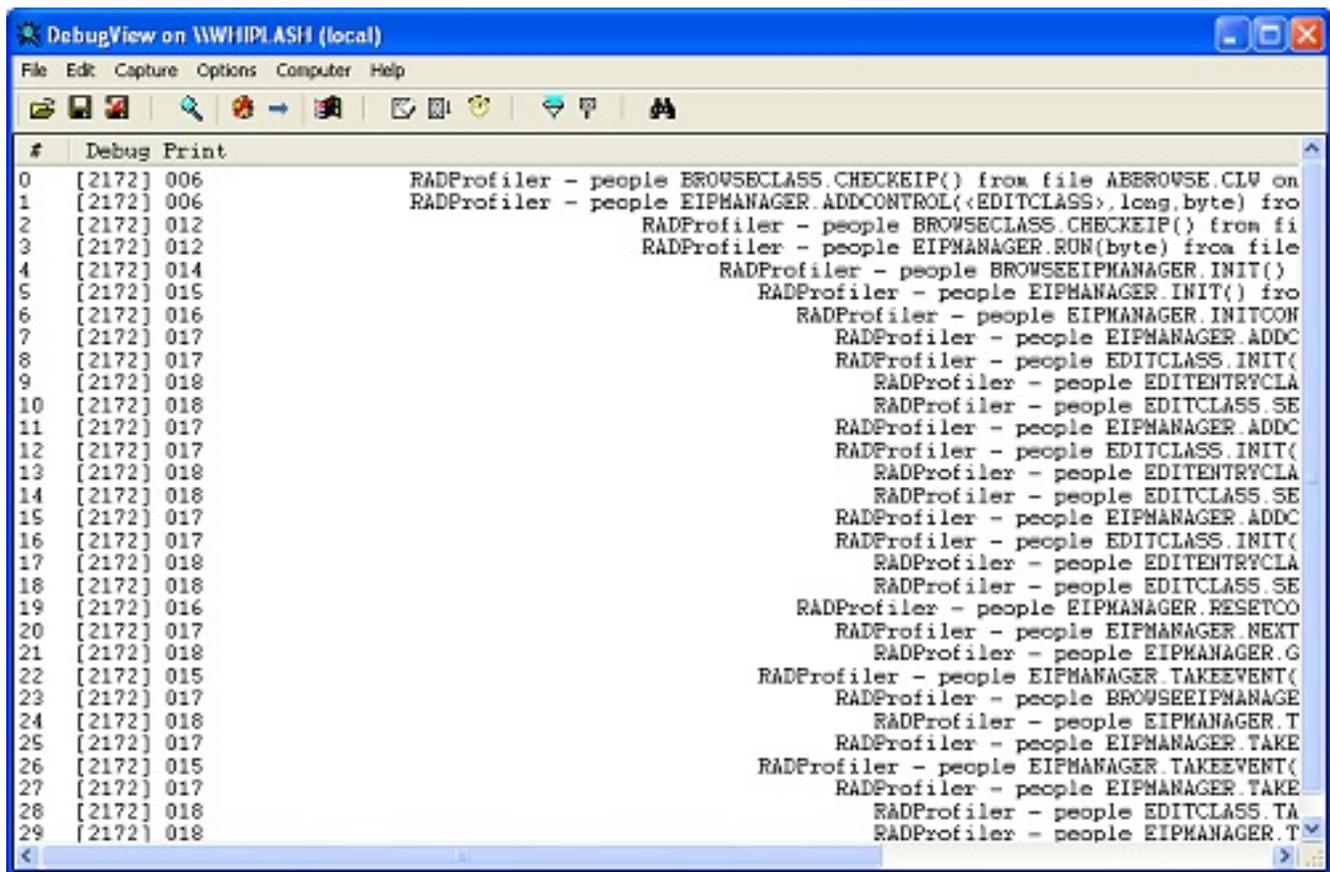


Figure 4. DebugView showing EIP calls

You can add your own trace calls to the log, if you wish, simply by calling the RADProfiler class's `Trace` method in an embed point. If you stick with the default class name, you'll need to write something like:

```
RADProfiler.Trace('my trace info here')
```

To keep typing to a minimum, I generally rename the profiler class instance to `pr` or `dbg`, in the global extension template's `Classes` tab.

DebugBreak

Another debugging option, available on the third template tab, is debug breaks, where by you specify a string on which the debugger should be invoked. RADProfiler does this by calling the `DebugBreak` function. The help file specifies that you need to be running the app from the debugger, and you also need to turn on debug breaks via the `Ctrl-Alt-D` hotkey from within your application. Actually I had some trouble getting this feature to work, but then again the reason I first started using the profiler hooks was because the

debugger gave me fits. 'Nuff said.

Summary

If you've ever wanted to know what's going on under the hood of an ABC (or any other application), or if you've ever had to debug anything but the simplest of code, RADProfiler (accompanied by DebugView) is an essential tool. Get it. Now.

[Download RADProfiler](#)

[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

Generic DLLs The Template Way

by Bjarne Havnen

Published 2005-05-26

A couple of years ago I found myself in the exact same situation as Jeff Slarve described in [DLLs and Reusable Code: Divide and Simplify](#), and I started to make my own generic DLL.

I like to use the application generator. Not because I don't like to handcode, absolutely not, but because the .APP and .DCT gives me fewer files to back up, and all that trivial coding is left to the templates.

I quickly discarded using the ABC templates for my common DLL; it just couldn't be done without problems. On the other hand, the ABC templates weren't really necessary either, since my common DLL would just include small pieces of code that really didn't need any ABC. The Clarion templates covered most of my needs.

There are a few drawbacks with using the Clarion templates

1. The DLL will be slightly bigger than a handcoded project
2. It is riskier to compile an application in different versions of Clarion: an APP can be modified if the templates change, but source code isn't touched by the generator.
3. As Jeff pointed out, the export or libfile must be modified before using the DLL

The first problem doesn't matter to me. I think the benefit of being able to use a dictionary and templates evens that one out. The second I overcame by either write-protecting the application, or by saving to different names for each version of Clarion. I prefer the latter because it gives me the possibility of using both C5.5 and C6.1 programs in the same directory.

For some time I was happy with my DLL application, I added several new functions each day that I just imported from other applications. I could live with having to modify the export file once, since the workload was mainly on adding the functions. But the day I started to make new functions one at a time, I found my approach unworkable. For every new function I was caught by the third problem: I had to modify the EXP before I could use the DLL's LIB file in my applications. These modifications were time-consuming.

I got a tip that I should make a project instead, but at that point, Jeff's article wasn't written and I failed in the attempt, mainly because of the name mangling. Then I realized that somewhere within the template system there had to be a piece of code that already handled this, right?

Not only did I find the function to do it, namely the `LinkName()` function, but I discovered that I could completely override the generation of the export file.

If you examine the Application part of the Clarion template chain, you will see two things that are relevant to the task. One is a call to `%ConstructExportFile`, and the other is a hidden embed, `%AfterGeneratedApplication`. This embed is the last embed before the application is considered generated, and any code you generate here will replace/override anything previously generated.. Since the export file is already made at this point, I know I can replace it with my own version. I would like my template code to be pretty similar to the original, so this gives me the first piece of code:

```
#At(%AfterGeneratedApplication),Where(%ProgramExtension='DLL')
  #Insert(%AdiConstructExportFile)
#EndAT
```

Next I had to design the `%AdiConstructExportFile` group. What I did was simply copy the `%ConstructExportFile` group and remove anything that wasn't related to the procedures. All I need is the procedures to be exported, so why bother with the global data and files?

The essential code in this group is very small:

```
NAME %Application GUI
EXPORTS
  #FOR(%Procedure)
    #IF(%ProcedureExported)
      #SET(%ExpLineNumber,%ExpLineNumber + 1)
      %(LINKNAME(%Procedure & %Prototype)) @%ExpLineNumber
    #ENDIF
  #ENDFOR
```

Technically, I don't need all the conditions and embeds that are present in the complete code. However, I find it convenient to test the application as an executable before it is distributed, so while developing, the original files can be left untouched. Also, if it should come to a point where a definition is to be exported, embeds are there to override the override.

After reading Jeff's article I added another group to construct the includefile. The filename is set to `applicationname + inc`. Obviously, the code can be modified if this is not suitable.

```
#Group(%AdiConstructIncludeFile)
#Equate(%IncFile,%Application&'.inc')
#MESSAGE('Generating Module: %IncFile', 1) #! Post generation message
#MESSAGE('Generating Map Include File', 2) #! Post generation message
#MESSAGE(' ', 3)
```

```
#Create(%IncFile)
#For(%Procedure),Where(%ProcedureExported)
    %Procedure %ProtoType #<!%ProcedureDescription
#EndFor
#Close(%IncFile)
```

Thanks for the idea, Jeff. This creates the Map Include File for the module and saves me the trouble of writing up every procedure by hand in the template I use to activate my custom DLL. I now use a template to include the custom DLL as well, because I regularly forgot to remove the Standard Clarion DLL checkbox for the External module, and I use the same DLL compiled for two versions of Clarion.

The demo application (mystuff.app) has a few pretty cool functions you might want to explore, like the print functions, shellexecute wrappers, and AdiFilefind, which I wrote one day I couldn't find a file that I knew was on the disc. Also, DisplayQueue is a nice utility function I use when I need to examine and possibly modify the contents of any queue.

I also included a simple template (UseCustomDLL) that generates the necessary code to attach a custom DLL that has been made by the OverrideExport template.

[Download the source](#)

Reader Comments

[Add a comment](#)

- [» In your INC file you should add the DLL\(dll_mode\) or...](#)
- [» Thank you, Carl. I am looking forward to try that. I...](#)
- [» Source updated! My apologies, I'd neglected to include...](#)
- [» > Will adding the DLL attribute ensure that the procedure...](#)