

Clarion Magazine

Clarion News

- » [EZRound 1.6.1](#)
- » [KlariSoft PDF Template](#)
- » [SB5 Developer Edition Beta Build 1055](#)
- » [SQL Seminar/CHT Discount](#)
- » [Carl Barnes Vacation Schedule](#)
- » [Office Inside 1.67](#)
- » [BST 3.90](#)
- » [xTipHotKey Class 2.6](#)
- » [DOS Printer 10.20](#)
- » [xProgress v1.2](#)
- » [BST 3.901](#)
- » [Lindersoft Lightning Strike](#)
- » [New Product News From Tracker Software](#)
- » [Free EZRound Tutorials](#)
- » [Das Product Price Reduction](#)
- » [Das Registration Updated](#)
- » [EZRound Graphic Edge HTML Table Designer](#)
- » [CapeSoft Draw 2.13](#)
- » [NetTalk Source Code for Sale](#)
- » [CapeSoft HotDates 1.15 Beta](#)
- » [CapeSoft Email Server](#)
- » [xClarEditor 1.2](#)
- » [BG DICO 2.0](#)
- » [Clarion Template for EZTwain Pro](#)
- » [In-Memory Driver Now Available For Clarion 5.5](#)

- » [UltraTree.com Site Facelift](#)
- » [Paragon Changes Name To Enabling Simplicity](#)
- » [UK User Group Meeting March 14](#)
- » [Clarion/SQL Seminar Early Bird Special Ends Tuesday](#)
- » [cpTracker Source Code Available](#)
- » [Hotfix 9033 Posted](#)
- » [ClassViewer 6.0](#)
- » [ABC Free Templates and Tools](#)
- » [Fomin Report Builder 2.92](#)
- » [Updated C6.x Help Files](#)
- » [EasyExcel 3.07](#)
- » [Armadillo/Relevant Reach](#)
- » [Big Scheduler Tamer Template Suite 3.81](#)
- » [SoftVelocity To Back-Port In-Memory Driver To Clarion 5.5](#)
- » [Valutilities Free Stuff](#)
- » [yuFileTools 3.10](#)
- » [Process Binder Template Updated](#)
- » [EasyListPrint 1.11](#)
- » [xRegExp Class 1.0](#)
- » [Digital River To Remove Relevant Reach](#)
- » [dpQuery 2.03](#)
- » [Fiddler HTTP Debugger](#)
- » [POSitive To Provide Support For Nice Touch Solutions Products](#)
- » [IP Driver Demo App](#)
- » [Replicate Class Restructuring](#)
- » [Big Scheduler Tamer Template Suite 3.8](#)
- » [xTipOfDay 2.3](#)
- » [xWhatsNew Class 1.8](#)
- » [xRuntimeStyle Manager 1.9](#)
- » [ProDomus Clarion 6.1 Updates](#)

[\[More news\]](#)

Podcast



This first podcast of 2005 is the annual Planet Clarion predictions show. Andrew and Dave pontificate on Clarion 7, Clarion.NET, open source databases, the future of Microsoft, and the world according to Google. 00:44:43, 15724K

 [Listen now](#)

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

Latest Free Content

[ANN: Clarion Magazine's New E-Books and a Special Subscriber Offer](#)

E-books are another great way to get the information you want from Clarion Magazine. Your time is valuable; with our PDF e-books, you spend less time hunting down the information you need. We're collecting the best Clarion Magazine articles by top developers into themed PDFs, with the first releases in March. These e-books will be available for purchase by anyone, but we have a special deal for subscribers. Not only will subscribers get discounted prices and free updates on e-books, but everyone taking out or renewing a subscription between February 23 and March 5, 2005 will receive an e-coupon good for two free e-books!

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

Latest Subscriber Content

[PDF for January, 2005](#)

All Clarion Magazine articles for January, 2005 in PDF format.

Posted Tuesday, February 01, 2005

[When Edit-In-Place Bites Back Some More](#)

Henry Plotkin really thought he had a pretty clever solution to a problem when he submitted [When Edit-In-Place Bites Back](#) to Clarion Magazine. When a reader pointed out further problem with EIP and the use of the mouse, Henry went back to the drawing board.

Posted Friday, February 11, 2005

[**An Overview of the In-Memory Driver**](#)

Bruce Johnson continues his series on the new drivers from SoftVelocity with a look at the In-Memory driver, which works like any other file driver except that all data is stored only in memory, not on disk.

Posted Friday, February 11, 2005

[**An Introduction to CVS and WinCVS**](#)

Version control software has long been popular with software developers, and particularly development teams, because it offers a centralized repository for source code, and tracks and manages changes to that source code. But even if you're a software developer working on your own, version control software can be immensely useful, and not just for software, but for any file that changes over time. In this two part series, Nardus Swanevelder shows how to install and configure CVS and WinCVS, two popular and freely available open source products.

Posted Monday, February 14, 2005

[**An Introduction to CVS and WinCVS Part 2**](#)

Version control software has long been popular with software developers, and particularly development teams, because it offers a centralized repository for source code, and tracks and manages changes to that source code. But even if you're a software developer working on your own, version control software can be immensely useful, and not just for software, but for any file that changes over time. In this two part series, Nardus Swanevelder shows how to install and configure CVS and WinCVS, two popular and freely available open source products. Part 2 of 2.

Posted Friday, February 18, 2005

[**An Overview of the Dynamic Driver**](#)

Bruce Johnson continues his series on the new drivers from SoftVelocity with a look at the Dynamic Driver, which lets you create file structures at runtime. As Bruce shows, the real magic happens when you combine the Dynamic Driver with the In-Memory Driver.

Posted Friday, February 18, 2005

[**ANN: Clarion Magazine's New E-Books and a Special Subscriber Offer \(free article\)**](#)

E-books are another great way to get the information you want from Clarion Magazine. Your time is valuable; with our PDF e-books, you spend less time hunting down the information you need. We're collecting the best Clarion Magazine articles by top developers into themed PDFs, with the first releases in March. These e-books will be

available for purchase by anyone, but we have a special deal for subscribers. Not only will subscribers get discounted prices and free updates on e-books, but everyone taking out or renewing a subscription between February 23 and March 5, 2005 will receive an e-coupon good for two free e-books!

Posted Wednesday, February 23, 2005

[Converting TPS Files To A New Layout](#)

No matter how hard you plan your database design; sooner or later you will want to change the layout of your application's TPS data files. Clarion has a mechanism for converting files in the dictionary editor, but it doesn't work on files that contain BLOBs or arrays. It also doesn't help if your application is installed in multiple sites, where the Clarion dictionary editor is not available. If you make changes to the file's structure and then distribute your application, your end users will get that dreaded error 47. That is, unless you do what Abe Jimenez does.

Posted Thursday, February 24, 2005

[Notifying ClarioNet Clients With PushWindow](#)

ClarioNet is designed to so that the server responds only when the client makes a request, to minimize the amount of network traffic. This can be annoying when the server side needs to complete a batch process and you want to inform the user that everything is okay and the client has not frozen up. Jason Sylvestre shows how to solve this problem with ClarioNet's PushWindows.

Posted Friday, February 25, 2005

[PDF for February, 2005](#)

All Clarion Magazine articles for February, 2005 in PDF format.

Posted Monday, February 28, 2005

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

Printed Books & E-Books

[E-Books \(available March, 2005\)](#)

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](#)

ABC Templates and Tools Updated 02/25/2005

Modified the following class-based templates to use standard object-wrappers: System-wide Hot Keys, Handle Shutdown Requests, Key Sender, ThreadLimit, Lookup Auto-complete an entry field, and Registry. Various other changes/fixes.

Posted Monday, March 07, 2005

BST 3.904

Big Scheduler Tamer Template Suite 3.904 is now available. Includes bugfix in Wall Calendar, Weekly Planner, Booking; you can now use HotField Reget without using the update option.

Posted Monday, March 07, 2005

BST 3.903 Example Patch

An example patch for Big Scheduler Tamer Template Suite 3.903 is now available.

Posted Monday, March 07, 2005

vuAgent Special

vuAgent is now only \$19.00. Fully functional demo available.

Posted Monday, March 07, 2005

[vuFileTools 3.20](#)

Version 3.20 of vuFileTools is now available for download. This version adds 11 new functions (for 132 in all) including string manipulation, CPU make model and speed, and installed printer count and printer name functions.

Posted Monday, March 07, 2005

[Up & Up](#)

Up & Up is a new tool for processing and reporting multiple levels in a Clarion relational database. Features include: Autogeneration of multiple interacting full blown processes; Multilevel simplified totaling; Computed Fields; On the fly record sorting; Full ViewManager at every processing level; Support for the Clarion extended outputs (HTML, PDF, etc).

Posted Monday, March 07, 2005

[XPathManager 1.0](#)

xPathManager 1.0 is a class with a set of templates to enable your programs to work with data in different folders. Features include: You can use your own table for storing paths; You can use both "full" and "relative" path to your data files; XPathManager can work with multi tables TPS files; XPathManager can create missing path. No black box DLL, just source code. Supports single EXE, multi DLL (Local Mode, Standalone Mode), 32-bit. For Clarion 6.1, Clarion 5.5 and Clarion 5. Demo available. Price is \$69.

Posted Thursday, March 03, 2005

[BST 3.903](#)

Big Scheduler Tamer Template Suite 3.903 is now available.

Posted Thursday, March 03, 2005

[EZRound 1.6.1 XHTML Compatible](#)

EZRound 1.6.1 generates XHTML compatible code for the HTML containers. The update includes several new tutorials and many small enhancements that users requested. Registered users can use the "Check for Update" option on the help menu.

Posted Thursday, March 03, 2005

[EZRound 1.6.1](#)

EZRound 1.6.1 is available for immediate download. You can use your "Check for updates" option under the help menu to update your system. Changes include: Redesigned the program output and template files for XHTML compatibility; Added paste from clipboard buttons to HTML color controls; Fixed small bug where wallpaper change would not clear "No Wallpaper" message; Fixed problem where "Click Here to Visit the EZRound Website" was trying to go to and invalid URL since the page had been renamed to a PHP page; Fixed bug where a user could not select a file from the import menu and the design menu would be turned off; Removed redraw and slice of default image if user started to open a file then canceled the process; Improved the registration screen paste feature to automatically detect the name or key (in addition to when both parts are on the clipboard) and place them in the correct field; Changed table shape selector so that when you select a square shape that the corner size is automatically set to zero; Added context sensitive help to the general sections of EZRound; Added tutorials from the EZRound website to the distributed help files; Added a link to the help menu for the online tutorials at the EZRound website. The EZRound website has also had quite a lot of work with several new tutorials and resources added.

Posted Monday, February 28, 2005

[KlariSoft PDF Template](#)

Combining the power of the iSEDQuickPDF control and flexibility of the Icetips Previewer templates you can easily print/preview any PDF file in Clarion without Adobe. Demo available. Note: Icetips Previewer templates are required (<http://www.icetips.com/previewer.php>).

Posted Monday, February 28, 2005

[SB5 Developer Edition Beta Build 1055](#)

SB5 Developer Edition Beta Build 1055 is now available. You can click "Check for Updates..." from within your SB5 IDE to update to this release. The full install (code signed SB5 installer) is also available.

Posted Wednesday, February 23, 2005

[SQL Seminar/CHT Discount](#)

If you sign up for Shawn's SQL seminar, and you're already a CHT subscriber, you can re-new your still current subscription at half the normal price (\$99 instead of \$199).

You can renew at any time and will simply get 12 added months to your present subscription's anniversary date. If you've let your subscription lapse you can renew your lapsed subscription for \$150 instead of \$250. If you are not yet a CHT subscriber you can become one for \$150 off: (i.e. \$249 instead of \$399). Once you have your acknowledgement of acceptance from Shawn, send a copy of it via email to support@cwandy.com. For more info on the SQL seminar go to <http://www.issda.com/mssql.htm>

Posted Wednesday, February 23, 2005

Carl Barnes Vacation Schedule

Carl Barnes is on vacation until 2/26. He will process orders for Clarion Source Search and CW Assistant on his return.

Posted Wednesday, February 23, 2005

Office Inside 1.67

Capesoft's Office Inside 1.67 is now available for download.

Posted Wednesday, February 23, 2005

BST 3.90

Big Scheduler Tamer Template Suite 3.90 is now available. Changes include: Added split columns/fields for conflict appointments; Stabilized scroll to date feature(BSTMonth); Added code to all update buttons to ensure list was selected to avoid errorcodes; BSTMonth added embed points to select resource button; Added checkbox to use same icon for duration (regular and conditional); Greenbar now only colors the first row if it is a full hr; Added and subtracted few seconds in Schedulers code to get rid of funky appointments; Added checkbox to GetDate button to automatically become a JumpDate lookup feature when used with Navigation Calendar or bar; Fixed default tooltip; Moved/changed a lot of code to make it more efficient, less file hits; BSTSched and Bookings changed code to allow descending key in Resource file; Clear the ghost hotfield displayed on schedulers and bookings after appointments retrieval.

Posted Wednesday, February 23, 2005

xTipHotKey Class 2.6

SealSoft's xTipHotKey 2.6 is now available. This release fixes a bug with tooltip text

being truncated.

Posted Wednesday, February 23, 2005

DOS Printer 10.20

DOS Printer 10.20 is now available. DOS Printer can capture DOS output sent to lpt1 and print it. It does this by installing a modified generic printer, and using Jim Kane's classes (www.par2.com) to identify the print jobs, steal them and delete them from the queue. It also monitors for up to four different filenames which can be sent to four different printers. In this way, a DOS program can create a text file and send it to a specific printer just by giving it the correct name.

Posted Wednesday, February 23, 2005

xProgress v1.2

SealSoft's xProgress is a class and code/control templates to show the progress of a process. Features include standard progress bar with additional settings, "snake" and slider progress bars, progress bars with images, and animated progress bars. Price is \$59.

Posted Wednesday, February 23, 2005

BST 3.901

Bugfix: When using only one column for conflict handling, it would get set to 0 width. Released Ver 3.901 patch and updated installer.

Posted Wednesday, February 23, 2005

Lindersoft Lightning Strike

Last week a lightning strike near the house was enough to destroy four machines including the server. The machines were plugged into two layers of surge protectors. But even the best consumer surge protectors are useless in the face of a lightning strike. One machine was not plugged into the electric socket at the time. All hard drives and processors are toast! Lindersoft's ISP has installed a new Stinger ADSL access concentrator unit now and UPS delivered five boxes containing Dell computer equipment on Friday. Unfortunately, Lindersoft lost emails sent between 2/14/2005 8:42PM and 2/15/2005 4:53 PM.

Posted Monday, February 21, 2005

[New Product News From Tracker Software](#)

The Tracker Software Adobe Acrobat PDF file viewer is now an official product in development with an early Beta available during the spring. Three products are in development for release: Viewer (competes with Adobe Reader) - no editing functionality; Developers Viewer SDK - to allow developers to embed and control PDF viewing and output (printing etc); PDF View & Edit - Feature rich editing of PDF file content and more. A forum for this product is now open - input from interested developers is welcome.

Posted Monday, February 21, 2005

[Free EZRound Tutorials](#)

A new tutorial section is now online at the EZRound web site. One addition, "Understanding EZRound Table Basics," is a detailed tutorial on how the EZRound image slices work with a regular HTML table to create a "container" for your content. The tutorial also has a section that gives you an overview of how EZRound makes it simple for you to manage your image slice files among all the other content on your web site.

Posted Monday, February 21, 2005

[Das Product Price Reduction](#)

Prices have dropped on Das Security (now \$199.00), Das Registration (now \$199.00), Das QuickHelp (now \$99.00) and Das Tools (now \$149.00). Registered users can renew at lower prices as well.

Posted Thursday, February 17, 2005

[Das Registration Updated](#)

The latest release of Das Registration includes multiple language support. You are now able to customize all screen text displayed in the messages into your own language. Simply translate the default language file shipped with Das Registration into the language you require.

Posted Thursday, February 17, 2005

[EZRound Graphic Edge HTML Table Designer](#)

Charles Edmonds has a new Clarion-built tool for web page designers that creates HTML tables with rounded corners, drop shadows and other special effects. EZRound

creates the graphics that are used to visually enhance your web pages. Then it slices them into perfectly sized pieces, writes them to optimized files and generates HTML containers for them. You just add your content. The 14 day trial edition of EZRound is fully functional with the exception that it will not export your files. Introductory price is \$19.95.

Posted Thursday, February 17, 2005

CapeSoft Draw 2.13

The latest release of Draw adds a number of new features and revamped documentation. The new features include: Smooth (anti-aliased) fonts; New methods to draw objects such as spheres and shaded ellipses; New DrawHeader template for adding great looking headings to each window in your app. The DrawHeader template allows you to easily add great looking headings to each of your windows. The headings are fully customisable, so you can globally change the header colour and style to suit your needs. Another advantage of this is that your DrawHeader is now easily edited, so displaying record specific info, or translating your app into other languages is breeze. CapeSoft Draw allows you to easily create any graphical effect that you desire, with fast, flicker-free drawing and instantaneous refreshes. The Draw library provides a large number of functions that make creating even complex effects easy - from objects such as spheres and shaded shapes to multiple layers and transparency types. Cost: \$119

Posted Thursday, February 17, 2005

NetTalk Source Code for Sale

Source code most CapeSoft accessories will now be made available. The first is NetTalk; source sells for \$1995.

Posted Thursday, February 17, 2005

CapeSoft HotDates 1.15 Beta

One of the obvious omissions in the initial release of HotDates was the lack of a day scheduler. This release includes a Scheduler control template, which completes the HotDates family of date-related control templates. There'll be more releases following shortly with extra features and tweaks. Current features include: A 1 to 12 month Calendar control template - displaying color-coded date information; Print the calendar as you see it; Legend control styler, making coloration easy and effective; A one month 'Update' type calendar with more expanded detail (per day); A 1 to 100 day planner control template (with zoom in and out); A Day scheduler; A DatePicker control

template; Update your information directly from the calendars; ABC and Legacy application template support. Cost: \$239.

Posted Thursday, February 17, 2005

[CapeSoft Email Server](#)

v2.20 Gold of the Email Server has been released. Source code is available. This allows you to customize the email server, add your own functionality, as well as seeing first hand how to inter-operate the various NetTalk objects inside the Email Server. Cost: \$30; Source Code Cost: \$499.

Posted Thursday, February 17, 2005

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

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 Friday, February 25, 2005

Track	Start	Length	Size	Description

Download or stream (Free Access)	00:00:00	00:37:08	13057 K	Special guest Bob Zaukere, SoftVelocity's president, talks about Clarion 7, Clarion.NET, the new IDE, the subscription program, and much more.
Individual tracks				
Track 1 	00:00:00	00:03:27	1215K	Intro
Track 2 	00:03:27	00:03:30	1237K	Bob Zaukere, SoftVelocity's president - chit chat Arnor can skip if he wants.
Track 3 	00:06:58	00:05:08	1808K	The Clarion 7 IDE, which has changed significantly since DevCon 2004
Track 4 	00:12:06	00:01:33	549K	Using old versions of Clarion in the new IDE
Track 5 	00:13:40	00:03:49	1343K	The IDE as a separate entity, apart from any particular Clarion language version
Track 6 	00:17:28	00:01:37	569K	The subscription program
Track 7 	00:19:06	00:03:21	1183K	Clarion 7 and Clarion.NET - what comes first, and at what price?
Track 8 	00:22:27	00:14:40	5160K	C7 and .NET apps - ports, migrations, other issues, and one little hidden nugget...

Planet Clarion for January 31, 2005

Track	Start	Length	Size	Description
Download or stream (Free Access)	00:00:00	00:44:43	15724 K	This first podcast of 2005 is the annual Planet Clarion predictions show. Andrew and Dave pontificate on Clarion 7, Clarion.NET, open source databases, the future of Microsoft, and the world according to Google.
Individual tracks				
Track 1 	00:00:00	00:13:36	4786K	Introduction, and fearless predictions about Clarion 7, Clarion.NET, and DevCon 2005.
Track 2 	00:13:36	00:07:07	2503K	Operating systems and servers
Track 3 	00:20:43	00:08:30	2994K	The new browser war

Track 4 	00:29:13	00:03:27	1215K	As Google takes over the world
Track 5 	00:32:41	00:03:11	1124K	Databases, and what's up with Btrieve and PostgreSQL
Track 6 	00:35:53	00:02:34	908K	PostgreSQL, MySQL, and the new MySQL worm
Track 7 	00:38:28	00:03:27	1215K	Microsoft takes its eye off the ball
Track 8 	00:41:55	00:02:47	984K	ClarionMag enters its seventh year of publication

Planet Clarion for December 23, 2004

Track	Start	Length	Size	Description
Download or stream (Free Access)	00:00:00	00:36:14	16990 K	In this Eve of Christmas Eve edition of the Planet, Dave and Andrew talk at length about the IP driver with Capesoft's Bruce Johnson. Track list now available.
Individual tracks				
Track 1 	00:00:00	00:01:39	775K	Intro
Track 2 	00:01:39	00:03:58	1401K	Bruce Johnson and African languages
Track 3 	00:05:38	00:02:59	1050K	What is the IP driver really for?
Track 4 	00:08:37	00:02:22	834K	Using the IP driver over a WAN
Track 5 	00:10:59	00:04:01	1417K	The IP driver's killer feature
Track 6 	00:15:00	00:01:58	695K	The IP driver and bandwidth issues
Track 7 	00:16:59	00:01:02	366K	The five minute conversion
Track 8 	00:18:02	00:03:47	1333K	Filtering: server side vs client side
Track 9 	00:21:48	00:06:09	2164K	More on bandwidth, saturated LANs, and WANs

Track 10 S	00:27:58	00:05:25	1910K	IP driver vs SQL and other technologies, and some tips/tricks
Track 11 S	00:33:24	00:02:17	974K	Color theory redux, and closing comments

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.

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

Clarion Magazine's 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.

We expect the first e-books to be ready for download in March, 2005.

E-book format

All our e-books are unencrypted PDFs, and are formatted to the same standards as our [print books](#). All you need to read them is [Adobe Reader](#) or any other PDF viewer.

Free e-books

If you take out a Clarion Magazine [subscription or renewal](#) by March 31, 2005, you'll automatically receive an electronic coupon for a **free e-book!** You can verify your coupon by logging in to your [My ClarionMag page](#).

Free updates

We're constantly updating our e-books with the latest articles. All e-book purchases come with a three month free upgrade policy; if you're also a subscriber, you get

unlimited free updates, as long as you have an active subscription.

Subscriber discounts

Clarion Magazine subscribers can purchase e-books at discounted prices.

Pricing

Most e-books will be in the range of \$9.95 - \$19.95.

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

When Edit-In-Place Bites Back Some More

by Henry Plotkin

Published 2005-02-11

I really thought I had a pretty clever solution to a problem when I submitted [When Edit-In-Place Bites Back](#) to Clarion Magazine. The problem I thought I had solved is that when Edit In Place (EIP) is configured on a browse, the Insert, Change and Delete buttons continue to work even while inserting or changing another record.

Specifically, on a browse, where even though I turned off the ability to insert new records or delete existing records, a user can press the change button, begin editing a record and then press the change button again before accepting the field. When this happens, the new value appears in the browse, but embedded code doesn't execute.

To see what I mean, start the demo app (see the link at the end of this article). Select File | Standard EIP Browse. Press the Edit Quantity (change) button or press Enter to start editing mode. Change the value and press Tab or Enter to complete the field normally. You will see a message and an "M" will appear in the Scan In column. Now start editing another row, but instead of completing the field properly after changing the number, click the Edit Quantity button again. The browse updates, the new quantity shows, but the message doesn't appear and the Scan In flag is not set.

A message failing to pop up is no big deal. But in the real application, the embed skipped is the one where inventory is updated and that is a big deal.

My solution was to disable the change button (in the EIP Field Manager for the field embed) and re-enable it after the field is completed properly.

To see this, select the Saving the User from Himself menu option in the demo app. When you start to edit a field, the Edit Quantity button dims, as in Figure 1.

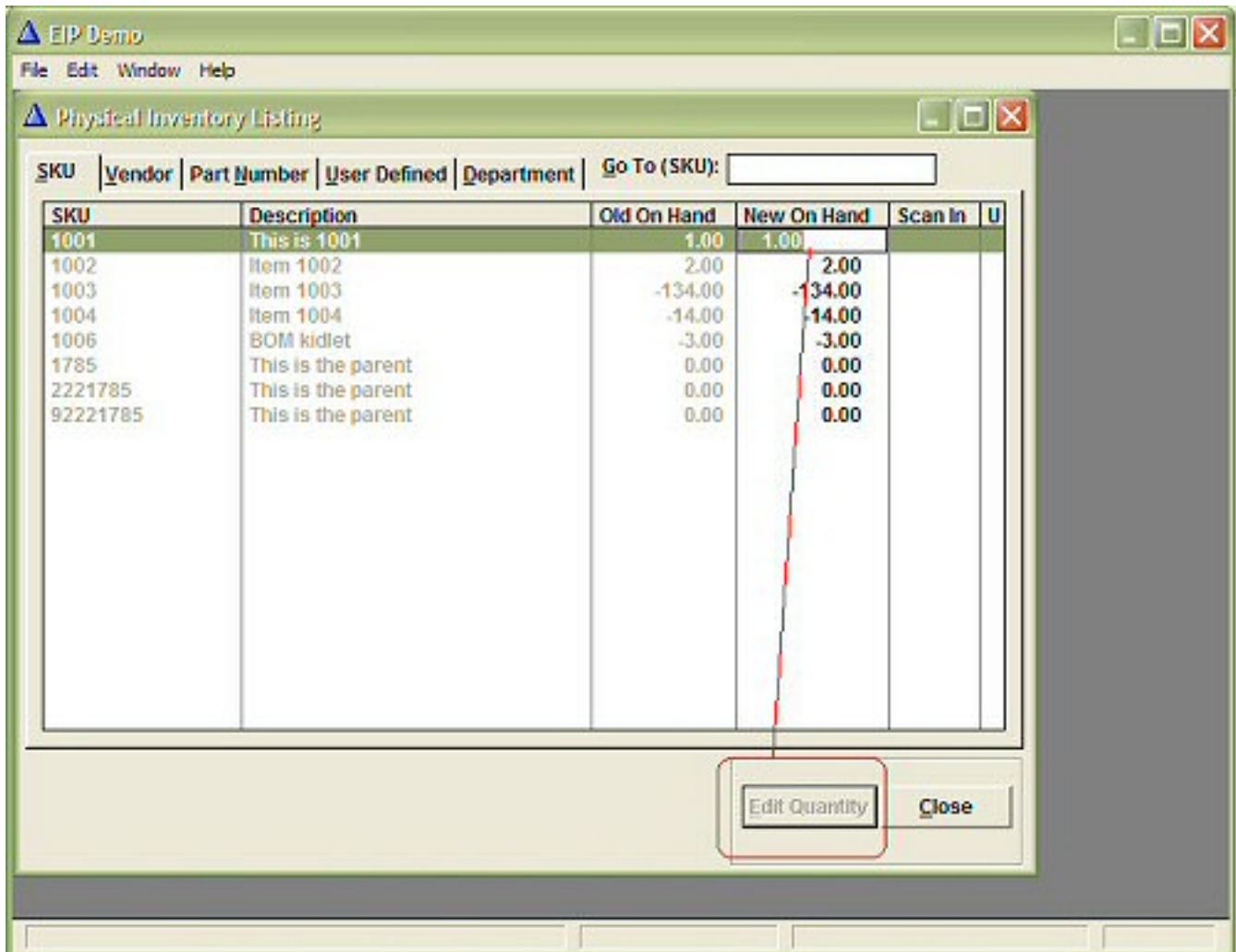


Figure 1. Button disabled during edit (view the [full sized image](#))

The user cannot press the Change button while changing a record (obviously, similar on-off switching can be added for any other button on the browse). Case solved. Or so I thought.

A new problem

Mr. Bjarne Havnen of ADITech Business Software advised me that my solution was interesting but does not quite solve the problem. He says my solution is not safe; there is still a risk of skipping embedded code, because my it addresses only part of the real problem. The real problem, as one of his users discovered not long before my article appeared, is use of the mouse. Clicking the mouse outside of the field being edited,

selecting another row or control, terminates EIP with just the consequences I started out to avoid.

To prove the point, he suggests starting my demo app and to begin editing a row. In mid-edit, select another row by clicking on it. EIP terminates, the new quantity displays and no embedded source code executes.

This is no different than pressing the change button in mid edit. Right back to where I started.

It seems, amongst other things, that Mr. Havnen's users are somewhat more creative than mine. It also seems that though I can provide a visual indicator, by disabling and dimming the Change button, that is neither sufficient to the need nor does it address the fundamental issue, which is change of focus.

Mr. Havnen suggests setting Action upon focus loss, an EIP option I never really paid attention to before:

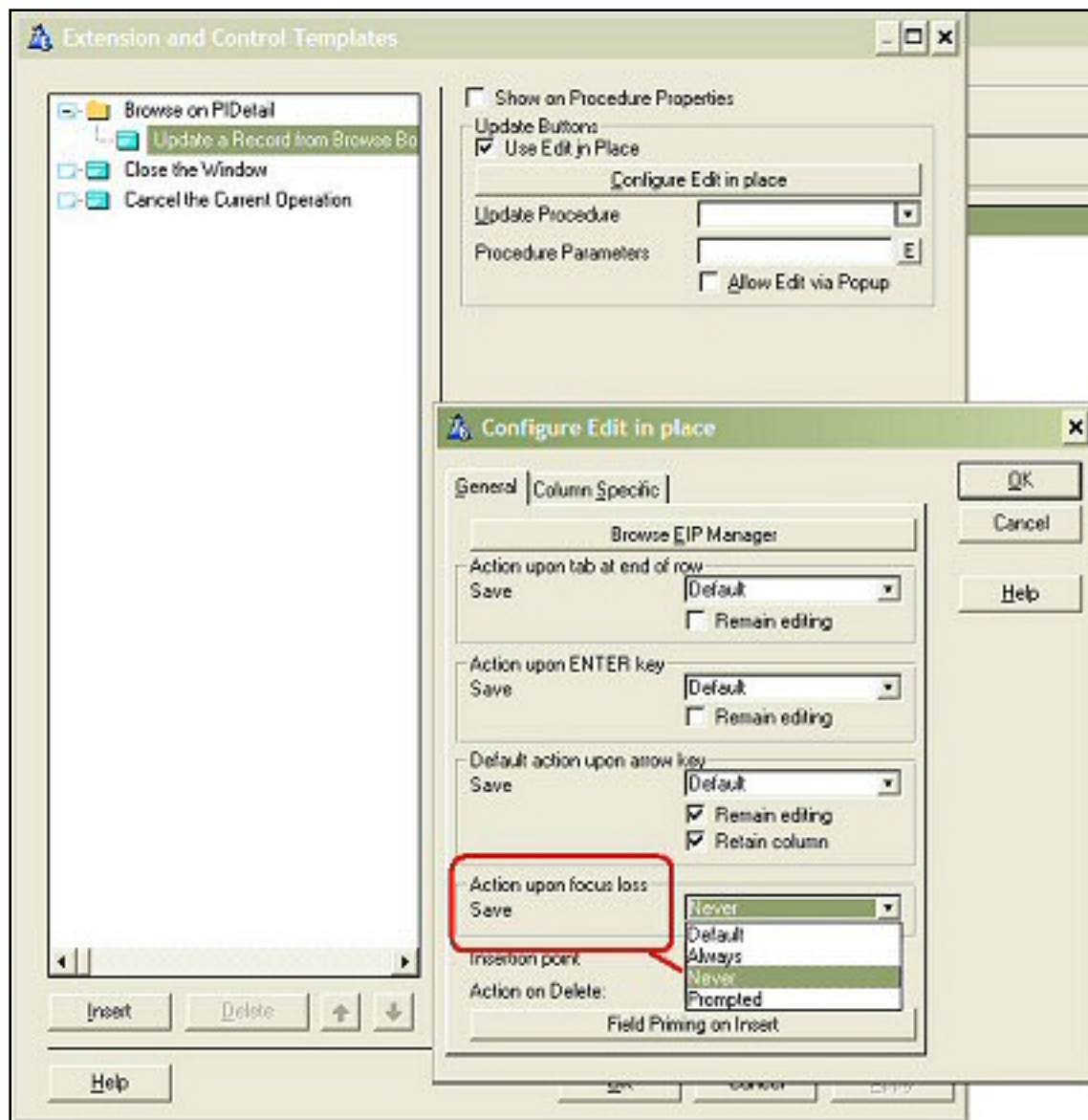


Figure 2. Action upon focus loss (view the [full sized image](#))

In the demo app, select the Safer Edit In Place menu option and do Mr. Havnen's experiment. Start editing an item then click on another. I set Action on focus loss to Never and because of that you will see that values revert to their initial value when you select another row. Mr. Havnen notes that, perhaps, this is not very user friendly but it is much safer.

I am not sure I agree with the "not user friendly" assessment. But, it is much safer, embeds cannot be skipped.

Another option

While looking through all this, I notice there is another option for Action upon focus loss that looks interesting, Prompted. When implemented, this option presents a standard "Do you want to save" type of message.

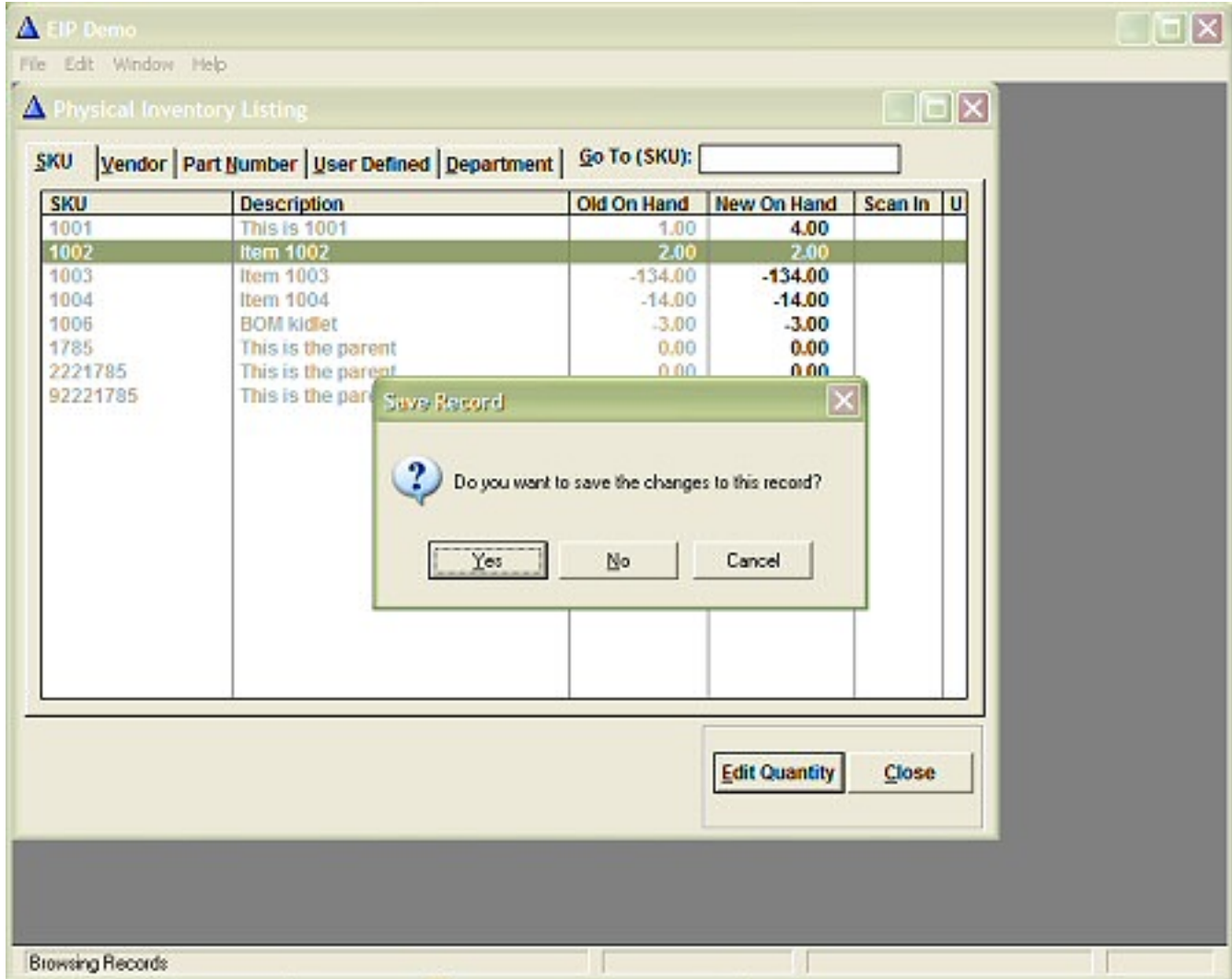


Figure 3. Prompted message (view the [full sized image](#))

Unfortunately, choosing to save changes from this dialog does not execute the embedded code, for the obvious reason that the field's TakeEvent method and the EditAction variable that must be tested are out of scope. Yet, if I recall correctly, browse objects have some standard behaviors that can be enlisted to my cause. (Truth to tell, I'm too close to the goal to give over now.)

My thinking is...

Specifically, browse update procedures are called from `ThisWindow.Run`

PROCEDURE (USHORT Number , BYTE Request). This method is called whether table update is done via EIP or a separate form procedure. ThisWindow.Run, in turn, calls ResetFromAsk to update/reset the browse object after touching a table, i.e. after an insert, change or delete.

ResetFromAsk is also a method where it is possible to test whether an action was completed and do some post processing. For example:

```
IF Request = DeleteRecord And Response = RequestCompleted
    Count -= 1
End
```

The only problem is that the field's TakeEvent method (the place where one tests EditAction and does post processing for EIP, i.e. the embed being used here) is out of scope and Response always returns 2, RequestCancelled (these results were obtained by trial and error). There is nothing to test!

However, it seems to me that if I can test whether or not the field's value has changed, I will have something testable to work with. In fact, this should be sufficient.

To provide a base against which to test, I save the field value in the browse's NewSelection method:

```
SAV:PID:Onhand = PID:NewOnHand
```

(SAV:PID:Onhand being a local variable created for this purpose). Then, in the ResetFromAsk, Before Parent call embed, I can do my test:

```
If PID:NewOnHand <> SAV:PID:Onhand
    PID:Scanned = 'M'
    Access:PIDetail.Update
    !get inventory record and update it
End
```

In the sample app, select the With Prompt menu option to see that deciding to save or not works the way it ought. By the way, the Cancel option returns you to editing. Now that's a treat.

I admit that I don't much care for Edit In Place. Perhaps that is why I'm unmotivated to dig further into the templates and classes, seeking further sophistication. My boss,

however, does, thinks it's just "very Windows."

I think it's a good thing I get paid by the hour.

[Download the source](#)

"hp" in fact prefers Hewlett-Packard printers but will use whatever is available. Born in New York City, hp is a self-taught Clarion developer doing a substantial amount of work for hospital gift shops.

Reader Comments

[Add a comment](#)

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

An Overview of the In-Memory Driver

by Bruce Johnson

Published 2005-02-11

The In-Memory Database Driver (IMDD) was the first of SoftVelocity's Driver offerings in 2004. Like many others I was an immediate adoptee. However some of the strengths of this driver have not become apparent until recently.

In its basic state the IMDD is a simple idea. It is a complete File Driver where the data is stored completely in RAM. When the program closes, the file disappears.

Why not just use queues?

Since the dawn of Clarion time, the Clarion language has been blessed with the Queue. This extremely powerful, and yet extremely elegant feature was one of the primary attractions for me when I started using Clarion.

In one sense the In-Memory driver delivers pretty much the same functionality as the Queue, but it has several big advantages. First, because the In-Memory structure is considered to be a table, it can be defined in the dictionary. This means that all templates that work on tables automatically work on In-Memory tables. Suddenly a task like reporting on a Queue (which is relatively hard) becomes trivial.

Another improvement over queues is the richness of the interface to the table. Functions like PREVIOUS and NEXT come easily when coding for tables, not so easily when coding for queues. Coupled with the FileManager objects created for you by

ABC, it feels very natural to write hand-code for In-Memory tables.

If that's not enough, In-Memory tables have another unique feature not matched by queues. They're thread safe. That means that you can use them across threads without having to worry about thread synchronization. That's all taken care of for you by the File Driver itself. Queues don't have this code built in – it has to be managed by you. While thread safety is almost a side-effect of the driver, it's a significant feature, especially when moving apps from Clarion 5.5 to Clarion 6. In C55 you don't need to worry about thread synchronization, but in C6 you do. By using the In-Memory driver as a replacement for your existing global queues you effectively side-step the synchronization issue.

At the risk of flogging this dead horse some more, it's also important to note that the IMDD allows for case sensitive, or insensitive, keys whereas a Queue is always case sensitive. Because Clarion programmers are typically not used to case sensitivity, when you encounter it in a queue then it can be a surprise – and surprises lead to bugs.

Like queues though, IMDD files “belong” to your program. Two instances of the same program running on the same machine (or different machines) cannot see each other's IMDD tables – each process has its own table. So you cannot use IMDD tables (or queues) to share information between programs.

Concurrency Issues

While the IMDD is thread safe, like any other driver, and indeed like queues, it cannot prevent normal contention from taking place. In other words, if two threads edit the same record, at the same time, and save it back, then ultimately only one record will survive. This is analogous to two users opening the same record in a form at the same time.

Out of the Box

When I was in school, and we did math problems, we used to get relatively few marks for the answer to a problem. Considered more important was the work that lead to the solution. Well, in one of my apps I implemented a feature that let the user see, in quite some detail, many lines of rules and results so they could better understand the answer.

When I first did it I thought nothing of making a TPS file, and then storing the calculations in that. (For reasons beyond the scope of this discussion, using a Queue was going to be hard work). Because the feature slowed down the calculation it could be turned on and off. When this TPS logging feature was on, it added about 30 seconds to each calculation. Since the users only investigated answers they considered to be

wrong, it was no big deal. The biggest hassle I had was in periodically emptying the table (at a time when no-one was accessing it.)

One of the first things I did with the In-Memory driver was change this temporary table to be "In-Memory". All I had to do was simply change the File Driver for the table, as set in the dictionary, and recompile. The benefits were immediate – it was a lot quicker than the TPS file (knocking more than 90% off the time taken to do the calculation), the data didn't go over the network (so reducing network load), and best of all the data disappeared all by itself when the program ended.

Nice, but not earth shattering.

Recently I had reason to knock a financial simulation together. In this case all the tables in the dictionary (bar one) are In-Memory tables. The program predicts seven years of cash flows based on a business model. On each day things happen, customers come and go, and so on. My In-Memory tables fly along wonderfully. Once the simulation is completed I can do all the regular things, browses, reports, graphs – the whole nine yards. I can also, if I like, dump the data to a physical file, or I can simply clean out the In-Memory tables, tweak some parameters, and run the simulation again.

Not bad, but I could have done the same thing using queues. Except that working with data tables (defined in the Dictionary) is much nicer. One thing I couldn't do with queues had to do with duplicate records. Because the In-Memory tables are tables, they can have keys, and these keys support DUP and PRIMARY and so on.

Again, slick, but not essential.

The short story was that I was finding the odd-job for the driver. It was fitting very nicely into my available tool-chest, but it wasn't exactly rocking my world. Its main strength was the sheer simplicity of the concept, its main weakness was the lack of a killer punch.

A few days later, however, I found out just how powerful it can be, and where its killer strength really lies. But before I can get there I need to digress for a bit, and talk about one more new driver. Stay tuned for the next article in this series.

The In-Memory Database Driver costs \$225 and requires Clarion 6 to operate. It is available online from www.clarionshop.com or direct from SoftVelocity at 800-270-4562.

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

Reader Comments

[Add a comment](#)

- [» We have a lot of developers requesting IMDD for their 5.5...](#)
- [» And today is 2/17/2005 Bob](#)
- [» Hello Bob, Any chance of the DFD becoming C5.5...](#)

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

An Introduction to CVS and WinCVS

by Nardus Swanevelder

Published 2005-02-14

Version control software has long been popular with software developers, and particularly development teams, because it offers a centralized repository for source code, and tracks and manages changes to that source code. But what if you're a Clarion programmer, working on your own? Why do you need Version Control? Have you ever lost a file due to corruption? Ever wished you had a copy of a previous version? Ever wished you did not delete that file? Ever wished you knew why you made that change again? Do you have different tracks of the same software? If you answered yes to any of these questions you need version control.

But Version Control software is expensive, isn't it?

Thanks to open source software there is a popular, free option called Concurrent Versions System (CVS). That takes care of the cost objection. (There is also another up and coming open source version control system called [Subversion](#), but that is a story for a future article.)

But isn't using version control difficult? Not really, and to prove that I will take you through the process of installing WinCVS, and show you how to start using it to control your software versions.

Before I continue, I want to emphasize that the purpose of this article is to introduce you to Version Control Systems, and *not* to explain Soft Velocity's implementation of its version control interface. You can use any version control system without using the built-in version control functionality in Clarion 6. The built-in version control functionality uses TXA files for the modules in your application and TXD files for your dictionary. As far as I am aware

the checkout command in Clarion 6.1 does have a problem if you are using CVS as your version control system.

The method I am going to explain in this article will use the CLW files, INC files, APP files etc. The only catch using this method is that you have to import the APP and DCT files as binary files (more on this later). The advantage of using this method of version control is that it is compatible with all versions of Clarion. Yes, even Clarion 2.1 for DOS (although I haven't done this personally). And of course you can use version control with all sorts of files, not just Clarion files.

What is CVS?

CVS is a version control system, dating back to 1986, that has been developed in the public domain by many people. Currently, CVS is maintained as an open source development project hosted by [Collab.Net, Inc.](#) Links to the CVS source code, binaries for various platforms and documentation can be found on the CVS project [home page](#).

The biggest limitation of CVS is that it uses a command line interface. Since most developers prefer a graphical user interface, several groups around the world have developed graphical front ends to the CVS core. One of the better front ends that is available for the Windows Operating System is [WinCVS](#), another open source project.

WinCVS is a Windows GUI that runs on top of CVS. WinCVS also make use of CVSNT and Python, which I'll touch on later. For more information on WinCVS you can visit their [official web site](#).

To download the latest version of WinCVS go to the [official download page](#). At the time of writing this article the latest recommended version of WinCVS was WinCVS 1.3.20.3 (released 2004-11-21). To get the latest WinCVS documentation you can visit the following [download page](#). You will find WinCVS documentation in MS Word or PDF format, along with a FAQ, CVS documentation, presentations etc.

The WinCVS installation contains a version of CVSNT, which is the actual CVS server, but *if* you need to get a more up to date version of CVSNT you can visit their [official web site](#).

CVSNT is the most widely supported, feature rich and secure CVS Server and Client available today on Windows, Mac OS X, Solaris, HPUNIX, and Linux.

CVSNT development began as simply a port of CVS 1.10 and then 1.11 to Windows, however it quickly became clear that vital new security and features were being added to CVSNT that CVS users on all platforms required. CVSNT then became available for Unix as well as Windows, and now supports most of the active software development platforms available.

CVSNT does install two services on your Windows machine. They are called "CVSNT Service" and "CVSNT Locking Service". You do not have to do anything with these services; they are started automatically when you switch your machine on.

Before you can start installing WinCVS you may need to download another application called Python. Python is an *interpreted, interactive, object-oriented* programming language. It is often compared to Tcl, Perl, Scheme or Java. Don't worry, you do not have to learn a new language - WinCVS uses Python but you won't need to write any Python code.

To get more information on Python visit their [official web site](#).

To download the latest version of Python go to the [download page](#). Remember to download the Windows version. At the time of writing this article the latest recommended version of Python was the Python 2.4 Windows installer. I had problems installing and running this version and reverted back to Python version 2.3.4

It is just about time to start the WinCVS installation process, but before you do that there is one more tool you might want to use. WinCVS supports the use of an external file comparison tool, and thanks to open source you have a couple of free options available to you:

CSDiff is a basic file comparison application and is available at the following [web site](#).

KDiff3 has more features than CSDiff but is more difficult to use, you can find it at the following [web site](#).

After you have downloaded the above applications you can start the installation process:

Installing WinCVS

1. Unzip WinCVS13B20-3.zip
2. Run Setup.Exe
3. Follow the prompts and choose Full Installation
4. Follow the rest of the prompts and click on Install button

5. After WinCVS has been installed make sure that you tick Install CVSNT and click on the Finish button
6. Follow the prompts and choose Full Installation
7. Follow the rest of the prompts and click on Install button
8. Click on Finish button

The next step is to install the Python application

Installing Python

1. Run Python-2.3.4.exe
2. Follow the prompts
3. I changed the default location of Python to: c:\Program Files\GNU\Python
4. Click the Finish button

That is it for the installation of WinCVS, all you have to do now is to install your file comparison tool of choice.

Terminology

With any new subject there is always new terminology that you need to understand before you will be able to grasp the full concept of the new subject. Before I continue with the article it is necessary to explain a few points.

Repository

The CVS repository stores a complete copy of all the files and directories which are under version control. Normally, you never access any of the files in the repository directly. Instead, you use CVS commands to get your own copy of the files into a working directory, and then you work on that copy.

The repository can be local or can reside on a remote server. Note that the repository is not a subdirectory of the working directory, or vice versa; they should be in separate locations. For this article I will assume that the repository is *local*.

Working files

Working files means the place (folder) where you check your files out to, so that you can work with them. When you are finished with your changes you check (commit) them back

into the repository.

Modules

Modules are either symbolic names for some collection of source directories and files, or paths to directories or files in the repository

CheckOut

When you checkout you create or update the working directory with copies of the source files specified by modules. You must use checkout before using most of the other CVS commands, since most of them operate on your working directory. If you use checkout it will create directories any missing directories.

Commit (CheckIn)

Use commit when you want to incorporate changes from your working files into the repository. Commit will only change the files in the repository that you have really changed.

Running WinCVS for the first time

After you have finished the installation of WinCVS you need to configure the WinCVS system, create a repository, and import your source code before you can using the version control functionality.

Configure Preferences

1. Start WinCVS
2. Click on Admin, Preferences
3. On the Globals Tab
 - a. Make sure "Checkout read-only" is not checked.
4. On the CVS Tab
 - a. Complete the path to the "Home" folder for e.g. c:\WinCVS
5. On the WinCVS tab
 - a. Complete the path to the "External diff" file comparison application, if you installed it.
 - b. Complete the path to the "Default editor"
 - c. Complete the path to the "TCL Dll"
6. On the Command Dialogs Tab
 - a. Skip dialogs (don't show dialog screens)

- i. Update
- ii. Diff
- iii. Graph
- iv. Log
- v. Annotate
- vi. Status
- vii. File action

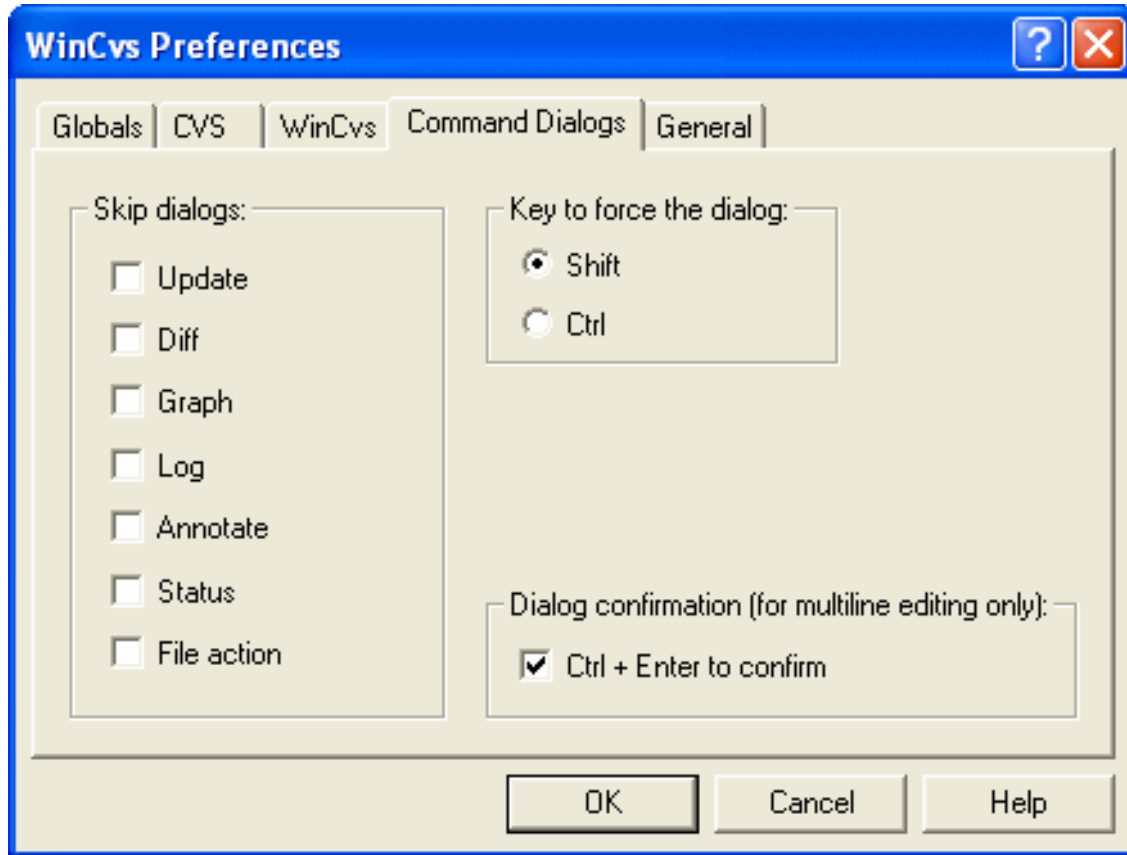


Figure 1. WinCVS Preferences

If you tick any of the "Skip dialogs:" options in Figure 1 you will not get the command dialog screen for that option when you execute it. To display the command dialog screen in future, you have to hold the Shift key in while choosing the function. The Update option's command dialog screen is displayed in Figure 2.

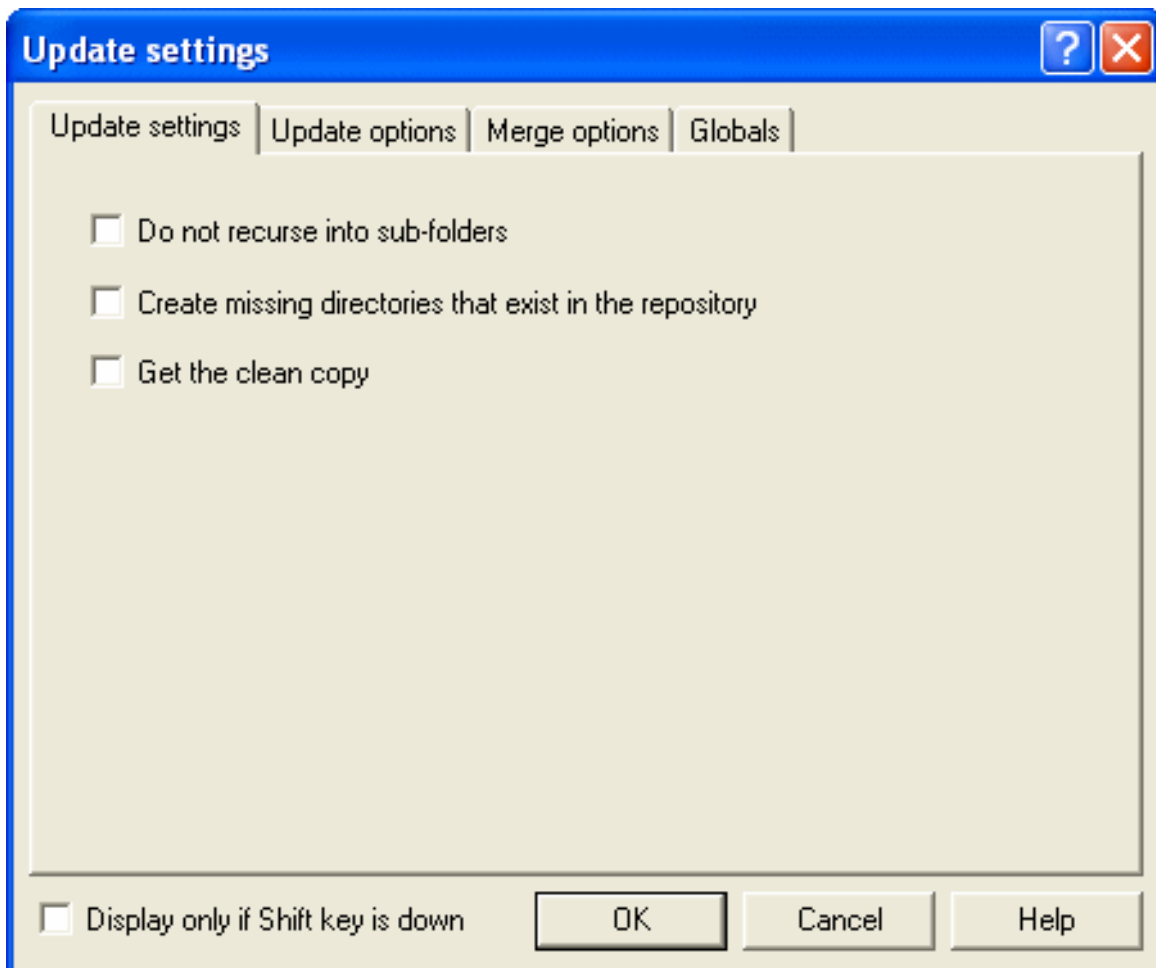


Figure 2. Update settings

Create repository

1. Click on Remote, Create a new repository
2. Complete the path to the "CVSRoot" and click on Ok. You can use c:\WinCVS for example.

First Import

1. Select the folder that contains the source files
2. Select the source files that you want to import into the repository
3. Click on Remote, Import Module. Figure 3 will display.

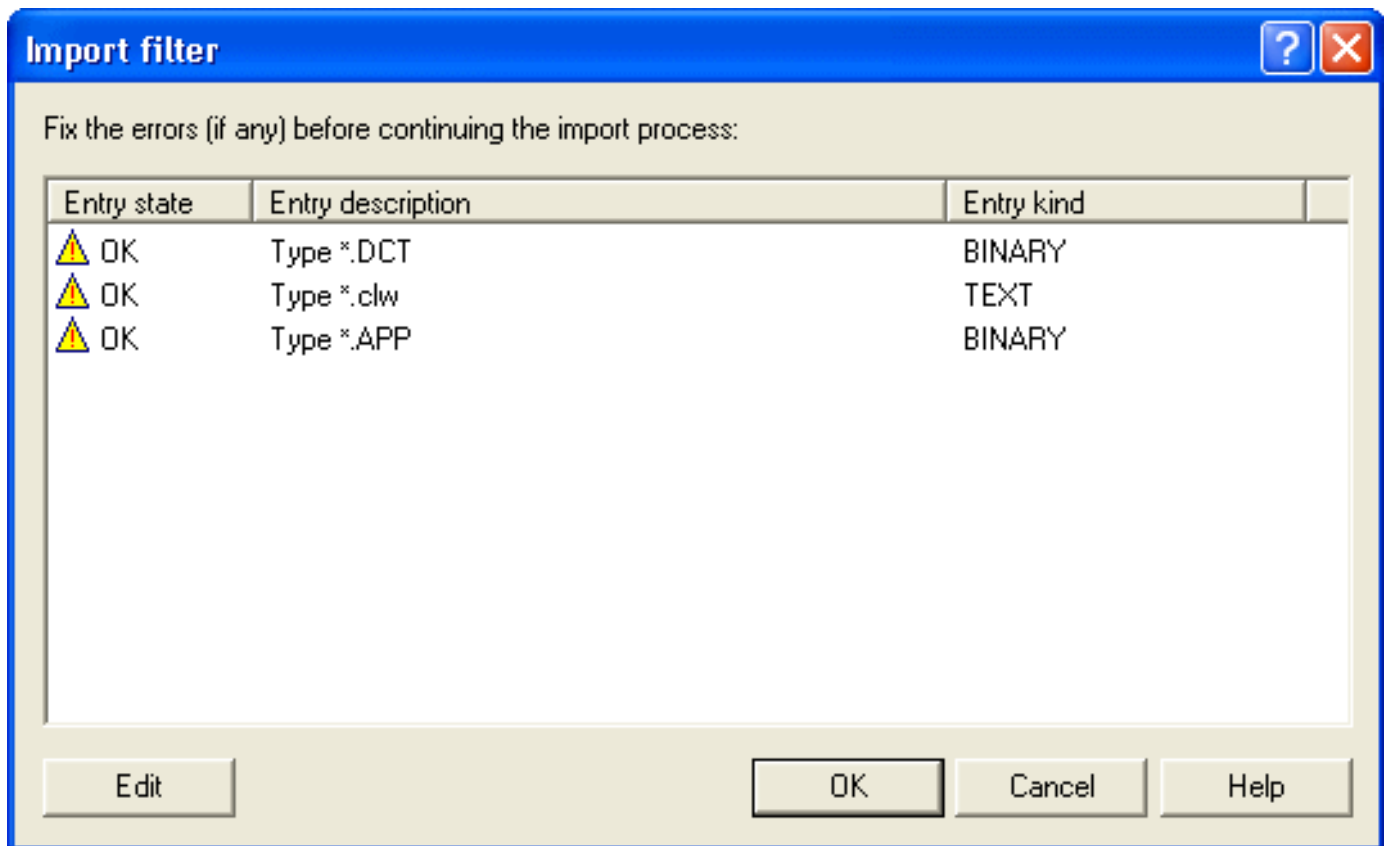


Figure 3. Import filters

4. Make sure that the Binary and Text files are indicated correctly
 - a. If you aren't using this method to import your files WinCVS will not import binary files as binary, unless you are using the right masks.
5. Click on continue, and the window in Figure 4 will display.

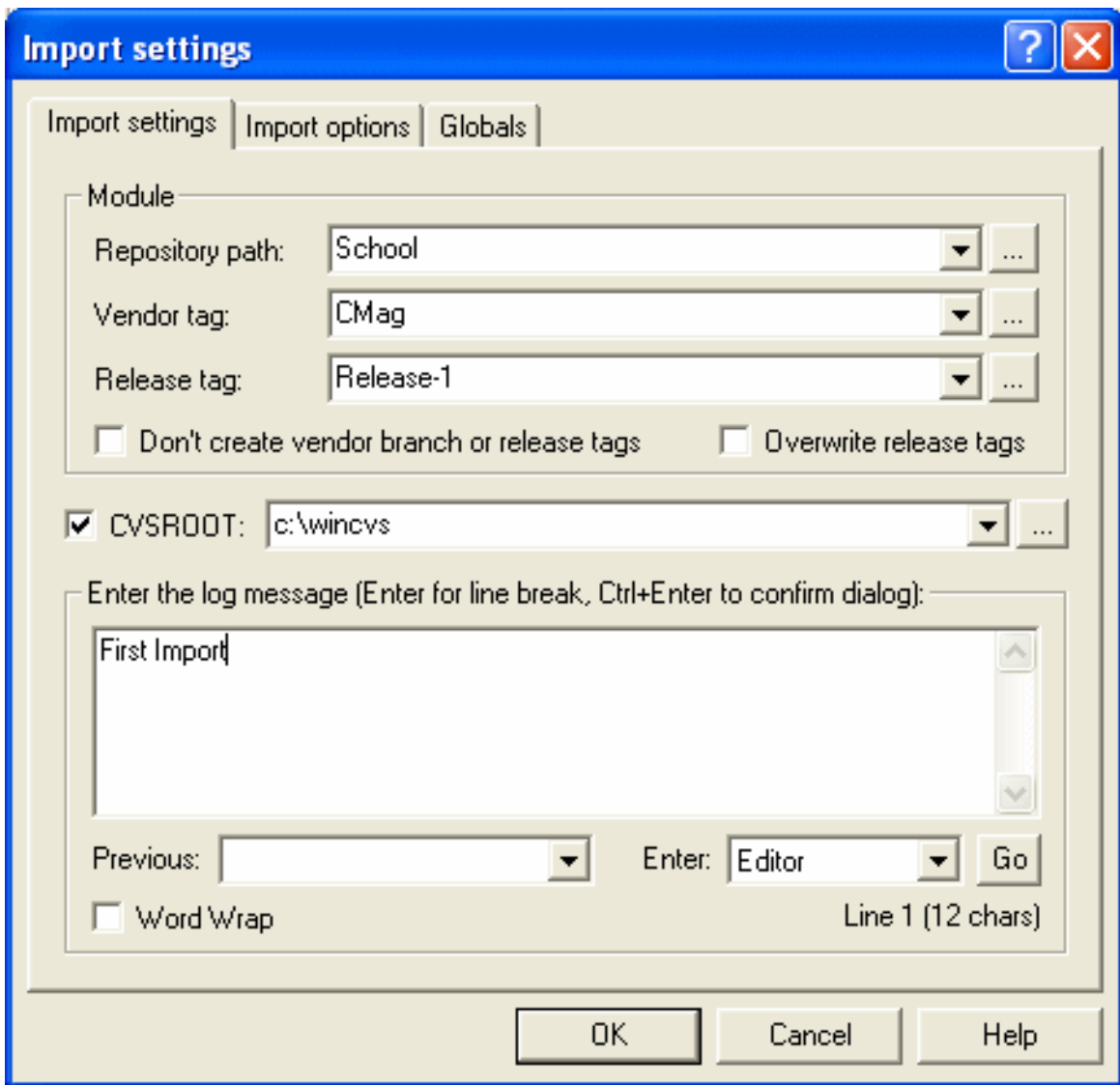


Figure 4. Import settings

6. Complete the following on the Import Settings Tab
 - a. Repository path - for e.g. choose an existing name or type a new name like "School". This will be your module name.
 - b. Vendor Tag - "Cmag" (The symbolic name is: branch 1.1.1)
 - c. ReleaseTag - Release -1 (The symbolic name is: - revision 1.1.1.1)
7. Leave the "Import options" tab the same
8. Change the Globals Tab
 - a. Uncheck Checkout read-only
9. Click on Ok

Figure 5 shows what the repository will look like after the import:

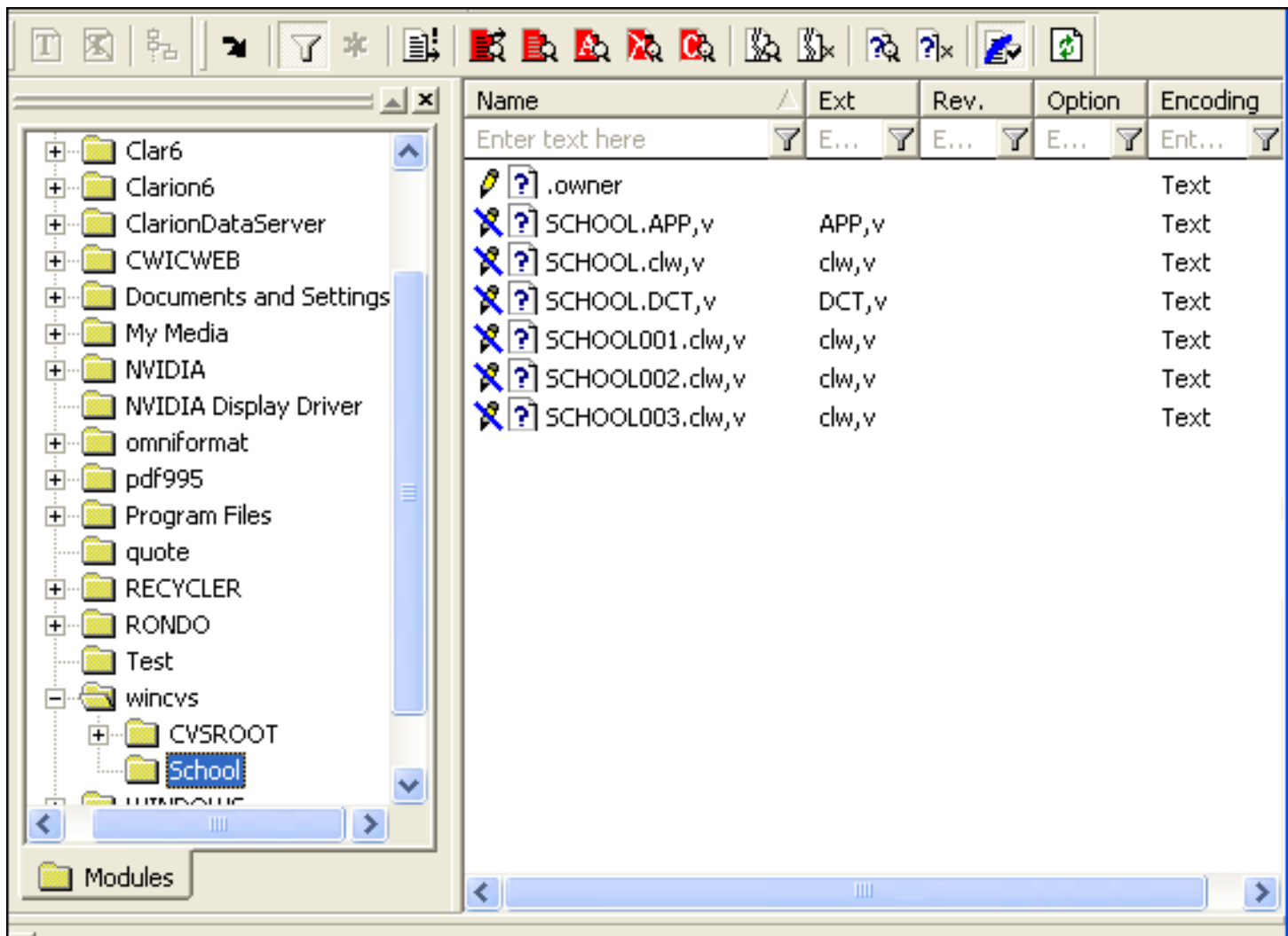


Figure 5. Repository after import

The ",v" after the file extension means that the file is marked as read-only. This is correct, so that the file set read-only. Also note that all files are marked as text. This is also correct, and the APP and DCT files will still be exported correctly as binary files.

You have your files in the repository and you are ready to start working on them.

Start with Checkout

Use the Checkout function before you make changes to your files:

1. Go to the Repository, and choose the Module that you created with the import - for e.g. School
2. Select all the files you want to Export
3. Click on Remote, Checkout Module
4. Complete the Checkout Settings Tab, as shown in Figure 6.

- a. Module Name and Path - School
- b. Local folder to checkout to: (Select the folder (one level up) where you want to export the files to)

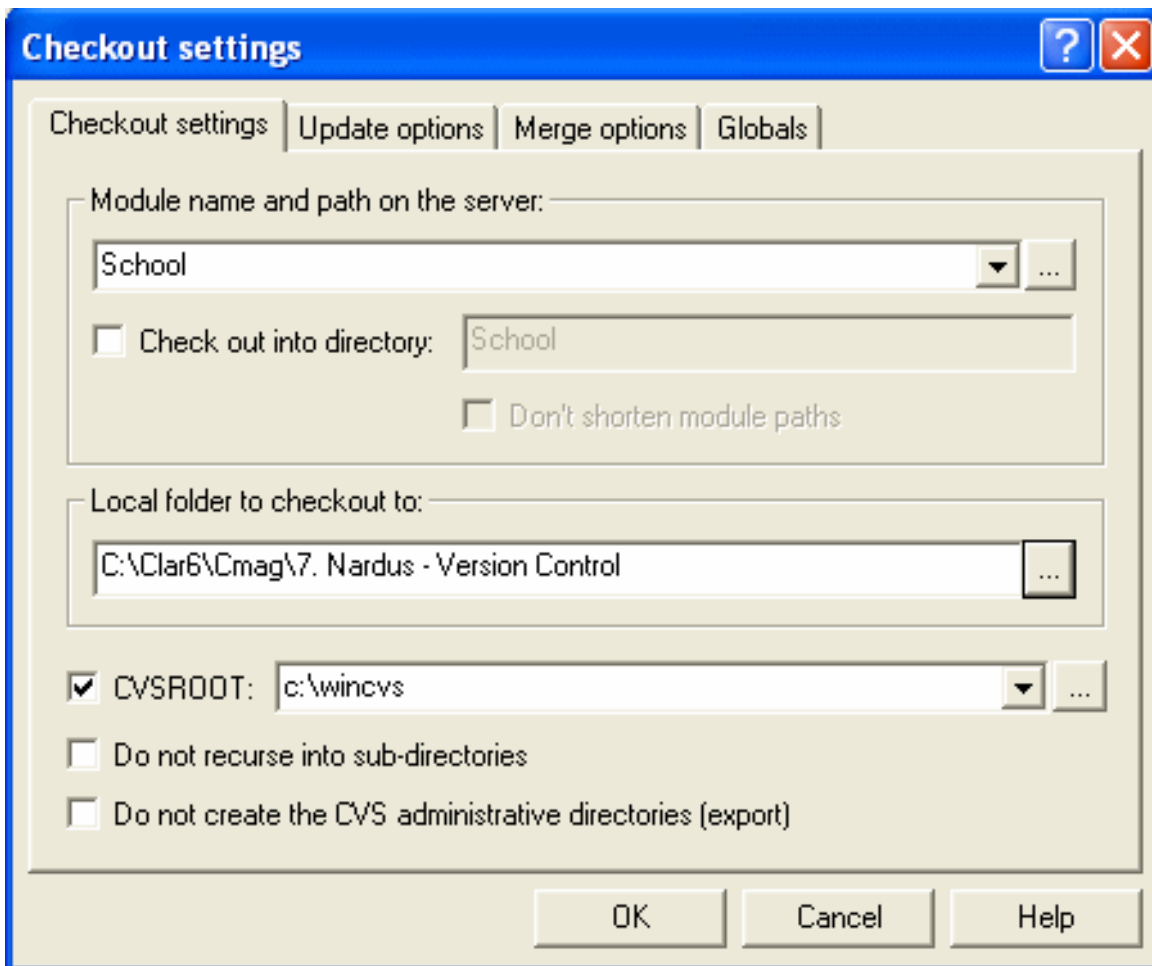


Figure 6. Checkout settings

5. Leave Update Options and Merge Options Tabs as is.
6. Make sure under the Global Tab that Checkout read-only is not checked
7. Click on Ok

Figure 7 is what the working folder will look like after the checkout.

Notice that the APP and DCT files were exported as binary files.

8. You are now ready to open Clarion and work on the files

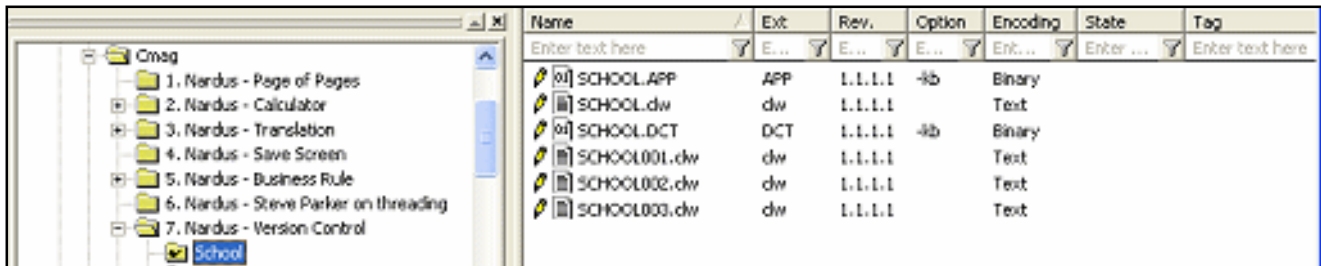


Figure 7. Working folder after checkout (view [full size image](#))

How do I know which files changed and which files are new?

Figure 8 shows how to distinguish between files that have been changed and files that have been added. New files will be indicated by a status of Unknown and changed files will be indicated by a status of Modified.

	01	SCHOOL.APP	APP	1.1.1.1	-kb	Binary	
		SCHOOL.dw	dw	1.1.1.1		Text	
	01	SCHOOL.DCT	DCT	1.1.1.1	-kb	Binary	
	?	SCHOOL001.BAK	BAK			Text	Unknown
		SCHOOL001.dw	dw	1.1.1.1		Text	Modified
		SCHOOL002.dw	dw	1.1.1.1		Text	
		SCHOOL003.dw	dw	1.1.1.1		Text	

Figure 8. Determining modified files/new files

Commit changed files

You need to commit changed files to update the repository

1. Go to your working folder
2. Select the changed files that you want to commit, right click and choose commit. You will see the window in Figure 9.
3. Enter a Comment that describes the changes you made. Leave the rest of the settings the same and click on OK.

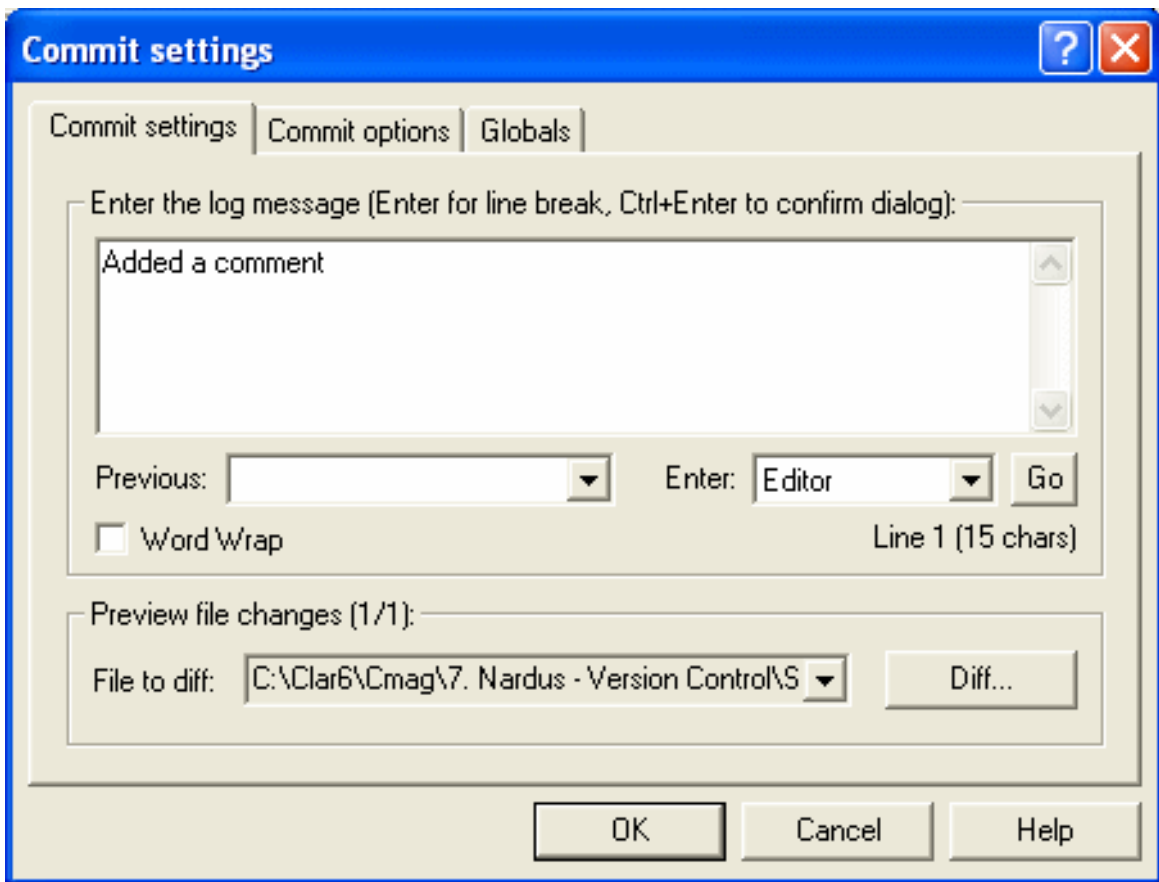


Figure 9. Commit settings

Commit new files

You need to do a commit to add new files to the repository

1. Select the new files that you want to add to the repository, click on the Add button or Add Binary button depending on the file type.
2. After you added the files to the repository you need to commit them to the repository. While the files are selected, right click and choose commit

Now that you have an idea of how to use the basic version control features, you're ready to start using WinCVS with Clarion. That's the subject of [Part 2](#).

[Nardus Swanevelder](#) was born and raised in South Africa. He was a networking engineer for seven years before he moved over to the commercial side of the business. Nardus has developed a Sale Cycle Management system for the Information and Communication Technology industry. He has been programming in Clarion since 1989, and holds B.Com and MBA degrees. In his spare time Nardus lectures Financial Management to B. Com Hons students at North-West University.

Reader Comments

[Add a comment](#)

- [» I think it is well worth mentioning the TortoiseCVS client...](#)
- [» Thanks for pointing that out. TortoiseCvs is another...](#)
- [» Can someone elaborate on "As far as I am aware the...](#)
- [» Shane, My tests show that a checkout from inside...](#)
- [» Gordon Yes, Tortoise is worth mentioning. And yes,...](#)

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

An Introduction to CVS and WinCVS Part 2

by **Nardus Swanevelder**

Published 2005-02-18

In [Part 1](#) I covered the basics of version control using CVS and WinCVS. Now it's time to look at how to apply these techniques to your Clarion development.

To begin with, you can use filters to reduce the number of files in your WinCVS view. As Figure 10 shows, you have Name, Extension, Revision, Option, Encoding, State, and Tag filters available to you.

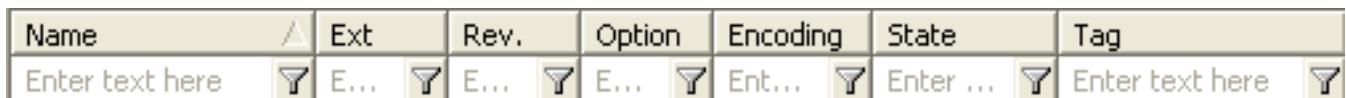


Figure 10. Filter settings

Filename Filter

.clw;.app - only display the CLW and APP files

Extension Filter

clw;app - another way to display only CLW and APP files

Revision Filter

Here you can enter a revision number like 1.1 for example. It will then display all the files with the revision number of 1.1.

State Filter

Modified - only displayed modified files

How do I check the differences between various versions?

It is easy to check for differences between different versions of the same document. Just follow the steps below:

1. Go to the Working folder or Repository folder
2. Select the file, right click and choose Diff Selection, as in Figure 11.
 - a. Choose the "compare type" you want (see Figure 12):
 - i. Local copy against the same remote revision
 - ii. Local copy against another revision/tag/branch or date
 - iii. Two revisions/tags/branches or dates against each other
 - b. If you want to use your external compare program make sure that you ticked "Use the external diff"

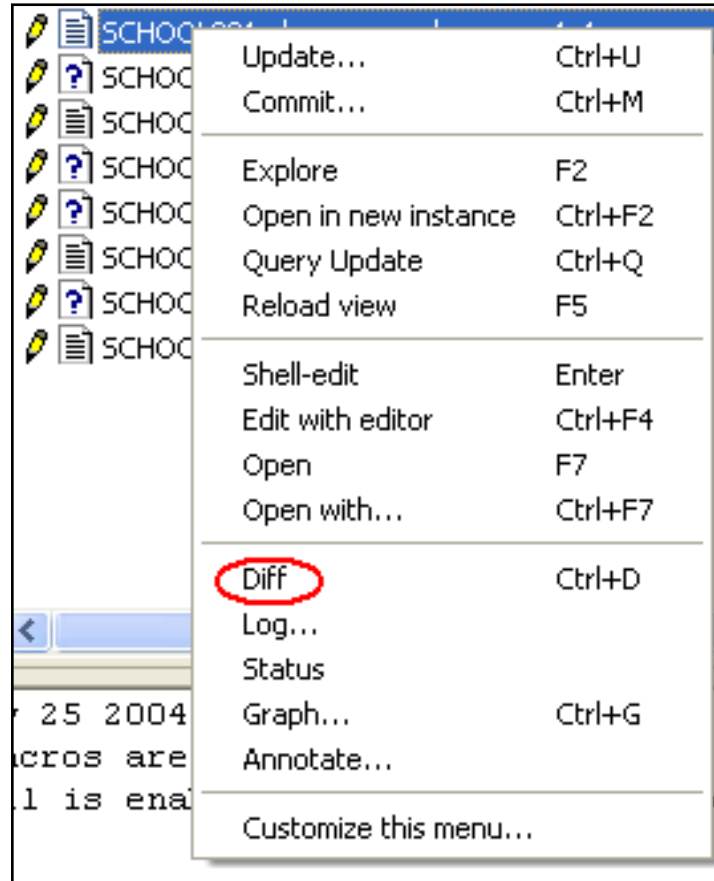
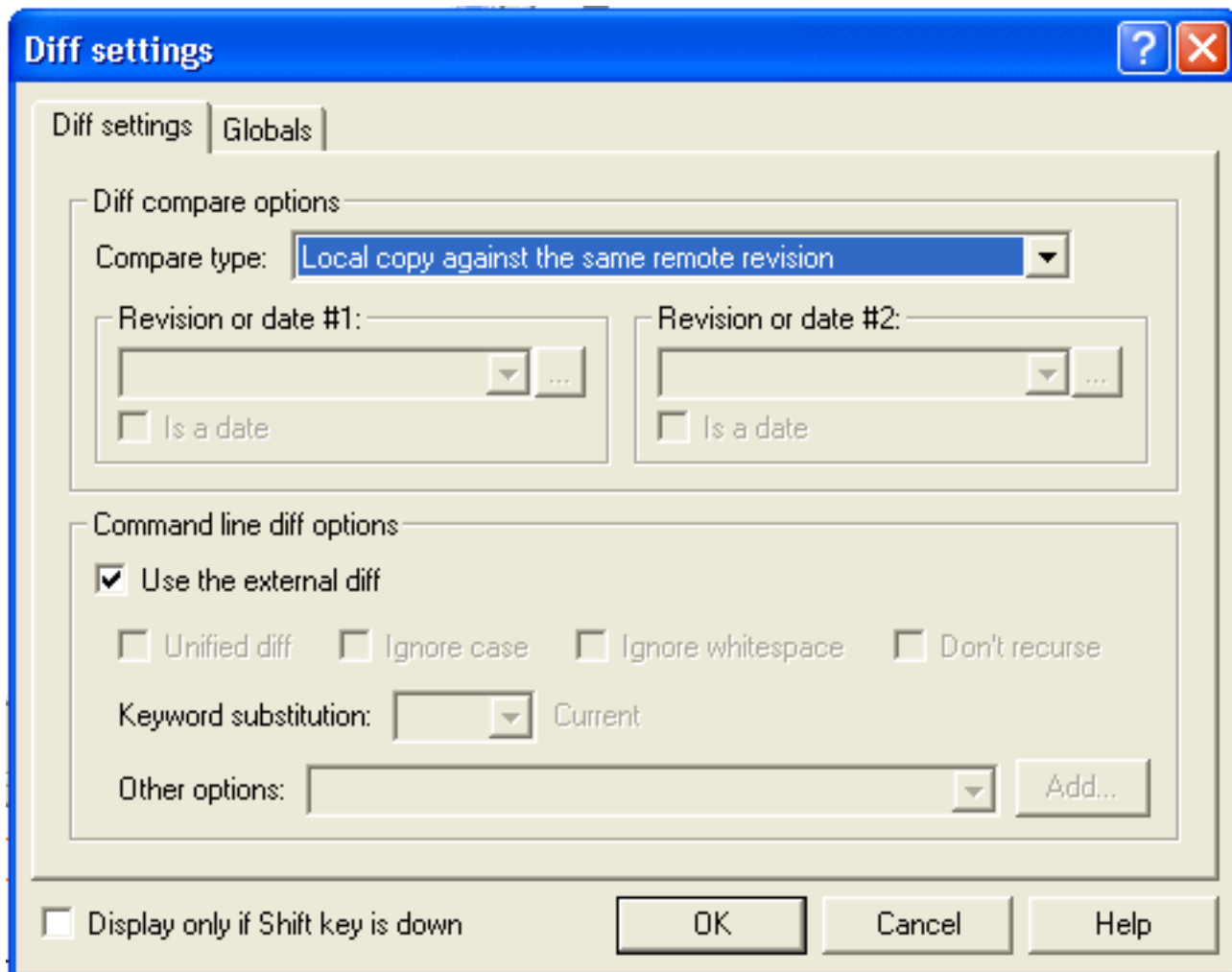


Figure 11. The Diff menu item**Figure 12. Using an external difference program****How do I get a Log of the changes that was made to a file?**

Getting a list of changes to a specific file is easy – just follow the steps below:

1. Go to the working folder
2. Select the file you want to have a log off
3. Right click, choose Log Selection
4. You will get a log in the WinCVS Output Pane. See Figure 13 for an example.

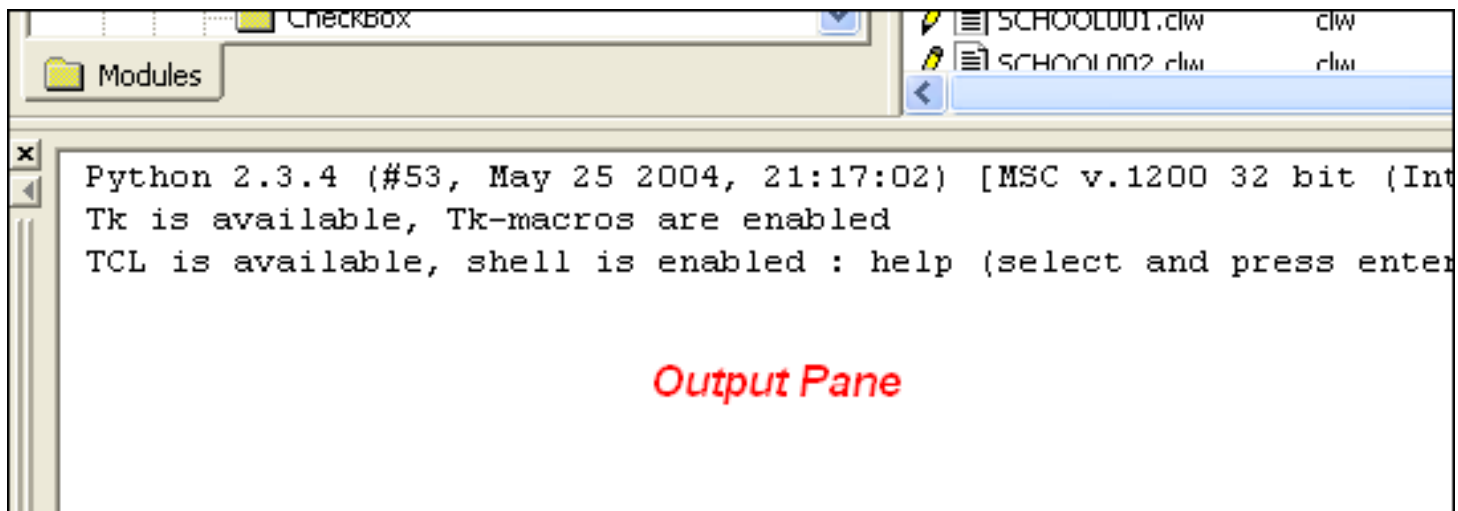


Figure 13. The output pane

5. The log will look something like this:

```
cvs log SCHOOL001.clw (in directory
  C:\Clar6\Cmag\7. Nardus - Version Control\School\
  ***** CVS exited normally with code 0 *****
  Rcs file : 'c:\WinCVS/School/SCHOOL001.clw,v'
```

```
Working file : 'SCHOOL001.clw'
Head revision : 1.2
```

```
Branch revision :
Locks : strict
Access :
Symbolic names :
  1.1.1.1 : 'Release-1'
  1.1.1 : 'CMag'
```

```
Keyword substitution : 'kv'
Total revisions : 3
Selected revisions : 3
Description :
```

```
-----
Revision : 1.2
```

```
Date : 2005/1/12 9:44:4
Author : 'Nardus.Swanevelder'
State : 'Exp'
Lines : +1 -1
Keyword : 'kv'
```

CommitID : '85041e4f1635ebf'

Description :

filename: SCHOOL001.clw;

Added a comment

Revision : 1.1

Date : 2005/1/12 9:34:49

Author : 'Nardus.Swanevelde'

State : 'Exp'

Lines : +0 0

Branches :

1.1.1

Description :

Initial revision

Revision : 1.1.1.1

Date : 2005/1/12 9:34:49

Author : 'Nardus.Swanevelde'

State : 'Exp'

Lines : +0 0

Description :

First Import

=====

How do I get the status of a file?

To get the status of a file is straightforward – just follow the next steps:

1. Go to the working folder
2. Select the file
3. Right click, choose Status Selection
4. You will get the status of the file in the WinCVS Output Pane
 - a. The Status can be:
 - i. Locally Modified
 - ii. Locally Added
 - iii. Up-to-date

- b. Check Working Revision
- c. Check Repository Revision

How do I get a graphical representation of the revisions of a file?

Getting a graphical representation is very simple. Just complete the steps below:

1. Go to working folder
2. Select the file
3. Right click, choose Graph Selection
4. The right pane changes and now shows a workflow type of graph, as in Figure 14.

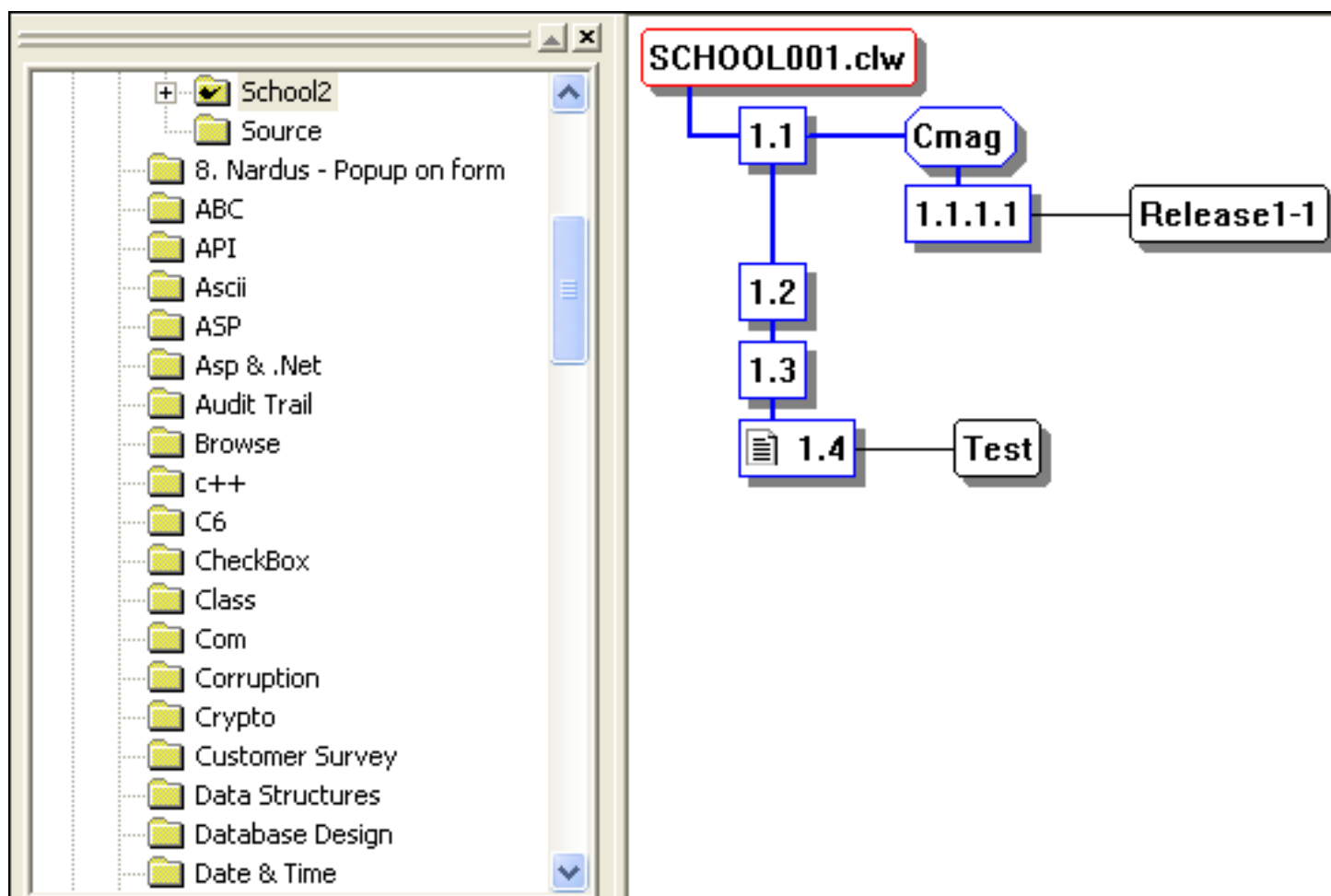


Figure 14. Viewing the revision graph

5. You can click on the different nodes which will display the following information in the Output Pane

Revision : **1.2**

```
Date : 2005/1/12 9:44:4
Author : 'Nardus.Swanevelde'
State : 'Exp'
Lines : +1 -1
Keyword : 'kv'
CommitID : '85041e4f1635ebf'
Description :
filename: SCHOOL001.clw;
Added a comment
```

- i. Revision – This is the revision of the file/node
 - ii. Date – This is the date and time when the changes were committed
 - iii. Author – Which person made the changes to the file
 - iv. State – Exp
 - v. Lines +1 -1
 - vi. Keyword
 1. kv – text
 2. b - binary
 - vii. Description – This is the description that you added the during the commit process.
 - viii. Filename – the name of the file that was committed.
6. If you want to retrieve a specific version right click on the correct revision and choose Retrieve or Retrieve as
 7. If you want to change the active node, right click and choose "Tag to selected (Sticky)". The tag will be indicated by a red box, as in Figure 15.

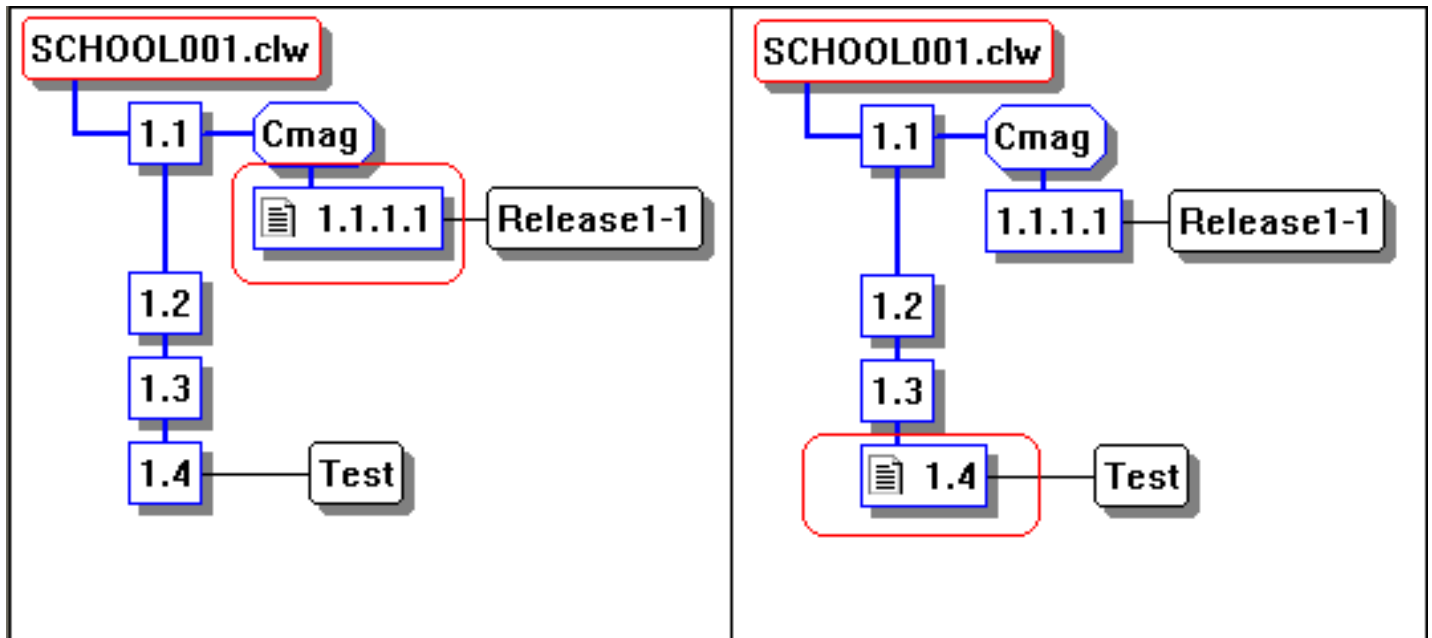


Figure 15. Changing the active node

8. If you right click and choose "Update (remove sticky)", your working folder will be updated with the version from the repository of the current selected node.
9. To close the graphical view click on the top node (School001.clw) in the tree and choose close.

How do I get a line by line representation of my file with version numbers on each line?

Getting a line by line representation of your file with version numbers on each line is a breeze – just follow these steps:

1. Go to working folder
2. Select the file
3. Right click, choose Annotate Selection
4. Remember that this will only work on a text file. It will not work on an application file as this is a binary file.
5. Here is an example pulled from the WinCVS Output Pane.

```

1.2 (Nardus.S 10-Jan-05): new comment by Nardus S
1.3 (Nardus.S 10-Jan-05): extra comment by Nardus
1.1 (Nardus.S 10-Jan-05):
1.1 (Nardus.S 10-Jan-05): MEMBER( 'SCHOOL.clw' )

```

Tips

Take note of the following tips; they will save you a lot of time.

1. Select the text that you need from the WinCVS output pane and cut and paste it into whatever application you want it. An example of the text that you can cut and paste is the log of changes you made to a file. You can cut and paste the log into MS Word and send it to your client as proof of the changes you made to the client's application.
2. Press the F2 key in the WinCVS application to open a Windows Explorer browser displaying the current selected folder in the WinCVS view.
3. If you are using a big application it will be beneficial to use one procedure per module. In Clarion, after you opened your application, click on the Application menu and choose "Repopulate Modules". Choose 1 procedure per module. In the School example this will increase your number of CLW files from 4 to 30 files. This can lead to longer compile times, but it lets CVS track procedures one at a time, which is very helpful when viewing differences. You must be careful, however, not to renumber modules. And that leads to tip #4.
4. Based on a tip from Rick Martin it will make life easier if you give your module names more meaning full names. Try using the procedure name as the module name. For e.g. instead of using School006.clw you can change the module name to UpdateCourses.clw.
5. If you are running a process in WinCVS and it looks like it is in a loop, it might be. To stop such a process you can click on the "Stop CVS" button. I got WinCVS to go into a loop when I used the import command incorrectly.
6. Read the WinCVS User Guide as listed under recommended reading, below. It is more useful than the WinCVS Online help.
7. It will be meaningless to use annotation and diff on a binary file.
8. You can import all Clarion bin, libsrc, template files as well as all third party files. This will enable you to go back to any version/patch of Clarion or your third party developer without re-installing any software. This tip is also from Rick Martin.

Summary

Version control software is an immensely useful tool for both single developers and development teams and CVS is the most widely used, free version control system currently available. With CVS, and the WinCVS client, you can easily make small

changes to source code and then undo those changes the next day. And because your work is versioned, if some incorrect change is made to the data somewhere along the line, just undo that change.

I've given you a basic background to CVS and WinCVS. As you progress in your knowledge, you may want to pay special attention to the topics of Tags, Branches, Watchers (multi-user environment) and you might want to learn how to use a remote repository.

Recommended reading:

- WinCVS User Guide – WinCVS1-.3.doc or WinCVS1-3.pdf (available at www.wincvs.org)
 - Version Management with CVS 1.12.9 by Cederqvist – cedeqvist-1.12.9.pdf (available at www.wincvs.org)
-

[Nardus Swanevelder](#) was born and raised in South Africa. He was a networking engineer for seven years before he moved over to the commercial side of the business. Nardus has developed a Sale Cycle Management system for the Information and Communication Technology industry. He has been programming in Clarion since 1989, and holds B.Com and MBA degrees. In his spare time Nardus lectures Financial Management to B. Com Hons students at North-West University.

Reader Comments

[Add a comment](#)

- [» Hi Nardus If you're editing the school.app and you modify...](#)
- [» Alan, I will let Nardus give his answer, but from my...](#)
- [» Bernard, Which paper are you referring to? Where can I...](#)
- [» Sorry I was in Mozambique for some Scuba diving. :-\] To...](#)

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the

[subscription agreement](#), is prohibited.

Clarion Magazine

An Overview of the Dynamic Driver

by Bruce Johnson

Published 2005-02-18

This is the third article in a series on the new drivers from SoftVelocity. Read about the [IP driver](#) and the [In-Memory driver](#).

The Dynamic File Driver was perhaps the most perplexing of drivers to me. To some extent, because we (at CapeSoft) have been generating file structures on the fly for many years in our File Manager products, I couldn't really see the point of it all.

In the past people have asked me about making some sort of generic product to handle dynamic file structures, but at the end of the day, while it may be a cool thing to do, it would have limited use in Clarion anyway. Because Clarion is template based (and because I'm a big believer in templates) dynamic structures just seem to fall out of the picture. Sure you can create a table, and sure you can loop through the table, but actually doing anything useful with the individual fields is a lot of hard work. Since, by definition, you don't know the structure at compile time, you can't actually *do* anything at compile time.

After discussing the concept for some time with SoftVelocity I was pretty sure this product was going to find small, important niches, but never really make it to widespread use.

Often the first step to seeing value is to accept that something can be done. Only later do you start seeing potential places where it could be used.

It took about a day for me to figure out that while the Dynamic Driver, and In-Memory Driver, apart from each other, are interesting, but not earth-shattering, together they are pure dynamite.

The main problem with the In-Memory Driver, in isolation, is getting data into and out of the table. There are a number of ways of doing this, but most are inelegant, or require you to have the same definition declared multiple times in your dictionary. The Dynamic Driver, however, has the ability to clone an existing structure, but change the driver. So by using both drivers you can have an In-Memory table and (say) a Topspeed table, with exactly the same structure, from a single dictionary definition.

Why is this dynamite? Well it allows a class to be written which can load a TPS file into a In-Memory file, or vice versa, with a single line of code. This class (and template), incidentally, ships with the Dynamic Driver; SoftVelocity calls this Global Caching.

Global Caching

The idea behind global caching is simple. Identify those tables in your app which seldom, if ever, change. Typically these have a few dozen records – maybe as many as a thousand. Lists of Departments, ZIP codes, Countries, that sort of thing. Looking at my dictionary this accounted for approximately one third of the tables there. These tables might be small, but they get used a lot, usually as a lookup or as a Select-From-List window.

So I add the Dynamic Driver global extension template to my app, and I identify these files. These files are now loaded when my program starts, and (optionally) saved again when the program ends. Since they almost never change I chose not to save them (and rather added a single line of Save code in the few places where they *could* be saved.

Other than this, *nothing* in my app changes. I don't need to change my dictionary to have these tables as In-Memory (although I can if I wanted to). I don't need to change my existing procedures.

Caching my global files immediately resulted in an overall improvement in the general "feel" of the app. There is nothing you can put your finger on, but overall it seems just that little bit more responsive.

Of course the benefits are not just simply that it goes quicker. It also it not hitting the server so often, and hence reduces the load on the server and the network. (This technique incidentally works just as well for SQL as for ISAM files. Indeed because of the one-off "get everything" request to the server you are making your client app very SQL efficient.)

This is very cool, but the good stuff doesn't stop there.

Database Design

As a good lawyer in a courtroom drama would say, "Stay with me, we're going somewhere with this..."

Like almost everyone working with databases I'm a big fan of third normal form. This is the idea that you should only store data in a single place on the server. So instead of typing in the Department description for each employee in a Department, I create a Department table, and link the two tables using a Department field in the Employee table. Unfortunately, third normal form has a flaw, which is best explained via an illustration.

Let's say a company has employees, who belong in departments. That's a many (employees) to one (department) relationship. But one employee can also belong to many departments. Think of a hotel where staff routinely work in both the Bar and the Restaurant. Now that many-to-one relationship becomes a many-to-many relationship. The database ends up looking something like this:

Employee table

Employee number

Employee name

Department table

Department Number

Department Name

EmpDeptLink table

Employee number

Department number

This is perfect third normal form. You can change the name of any employee in one

place, and you can change the name of any department in one place.

So what's the problem? The problem happens when the client says something like this:

"I want a report, with the departments listed alphabetically. Inside each department list the employees that belong to that department *in alphabetical order*"

It turns out that doing this is either not easy, or not efficient. That is to say it can easily be done, but it will be inefficient (and slow). Or it can be efficient, but (up to now) this means a lot of extra work. Since I'm a lazy git most of the time, and because optimizing reports doesn't have a lot of job satisfaction, I prefer the easy-but-inefficient approach.

For this report, this ends up looking something like this

```
Loop through Departments in alphabetical order
  Loop Through all employees in alphabetical order
    Do a lookup to see if this employee is in this department
```

A quick calculation on the back of a napkin shows that this will result in:

- d reads through the department table (d = number of records in the departments table)
- $e*d$ reads through the employee table (e = number of records in the employee table)
- $e*d$ reads on the link table

It sounds like a lot, and it is a lot. For 25 departments, 1000 employees, and say an average of two departments per person, this is 50100 disk reads. And what's worse it, escalates as a multiple of d , where d is the number of departments. For every lousy little extra department, this report is going get slower and slower.

And unfortunately it's not just the time needed to do the reads. This is a lot of network traffic, and an awful lot of processing for the server to do (especially if it is a SQL server.)

I *could* optimize it by using queues to cache the tables, and in some cases I have, but this is a pain. Can the Dynamic Driver, used with the In-Memory driver, give me some of the efficiency of queues without the pain? Yes it can, using *local* caching.

Local Caching

The idea with local caching is that a table can be cached for a single thread. Doing global caching of my employees file (for example) is not possible because employee details are changing all the time. So global caching is not an option. But if I can cache the Employee file, and the Link file *just* for the report thread, then I'm in business.

Consider this scenario. I can cache the department file globally. So that is taken care of before I even get to the report. So I only need to read all the employees into the In-Memory table, plus the Link records. Using the same numbers as above this results in 3000 reads. ($e + l$).

That's better than an order of magnitude improvement. Now all the data is in memory. My report still works the way it did before, but it's reading the memory tables, which are very fast.

Just thinking about it got me all excited. So I built some examples to test the theory. And I applied it to a real-world app. Frankly, fast reports still go fast. (I didn't bother caching those.) But slow ones see a performance gain of 80%-90%. That's staggering.

Bottom line: a report that took an hour to complete now prints in seven minutes. *That's* something the customer notices!

Conclusion of Part 3

History [famously records](#) that Steve Jobs was introduced to three new technologies on *that* day at Xerox PARC: OOP, Ethernet and the GUI. Since he saw the GUI first he was so gob-smacked by that that he missed the rest.

In some ways, coming so soon behind the IP Driver, the In-Memory/Dynamic combo has the danger of being a one-two punch. Both are unbelievably impressive, but you run the risk of missing one by concentrating on the other. Unlike the IP Driver though, which has limited appeal to SQL apps, the In-Memory driver, when used with the Dynamic File Driver is going to have a big impact on SQL applications. And all three drivers have a role to play when it comes to reducing network bandwidth.

Is this the end? Not by a long shot – I've just touched the surface of all three drivers. Next comes the really exciting stuff.

The Dynamic File Driver costs \$225 and requires Clarion 6 to operate. It is available online from www.clarionshop.com or direct from SoftVelocity at 800-270-4562.

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

Reader Comments

[Add a comment](#)

- [» Hi Bruce. Are you saying that with Global caching, you...](#)
- [» Hi Alan, You've almost got it. With global caching...](#)
- [» Hi Bruce Thanks for the write-up, which resulted in a...](#)
- [» Daan, answering that is slightly more than I can write...](#)

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

Converting TPS Files To A New Layout

by **Abe Jimenez**

Published 2005-02-24

No matter how hard you plan your database design; sooner or later you will want to change the layout of your application's TPS data files. Clarion has a mechanism for converting files in the dictionary editor, but it doesn't work on files that contain BLOBs or arrays. It also doesn't help if your application is installed in multiple sites, where the Clarion dictionary editor is not available. If you make changes to the file's structure and then distribute your application, your end users will get that dreaded error 47.

A number of people have tried to address this problem over the years, and there are some third party products available to aid in converting your TPS files to a new format. But I prefer to have control of my file conversions, so I code my own conversion routines. This article describes a technique I have used for years. I won't cover all of the angles - there are situations beyond the scope of this article, such as skipping versions and handling restored backups of old database versions - but the code I'll show is easy to implement, gives you full control of your conversions and takes care of the distribution issues.

Here is a list of the requirements or objectives this solution addresses.

Objectives:

1. You want a technique that works no matter what kind of change you want to make to a file.
2. You want to be able to distribute the application to customers or users who don't have Clarion or your dictionary file installed on their

computers.

3. You want the conversion of the files in your customer's site to run without the need for technical people.
4. The conversion must only reformat the data once, even if the user runs the program multiple times. This will enable you to integrate the conversion into your setup program, and make it run only once in an environment where various users share a database..
5. If the conversion program fails, you want to be able to restart it.
6. If somehow your users find a way to mess things up, you want to be able to recover and not lose any data.
7. You don't want your users to get error 47 (or any other error for that matter).

The only way I can make sure that my first objective is met is to create a new file with the attributes I want and then copy the data from the old version of the file into it. The challenge is in making your application accept the new file as if it was the old one. You really don't want to have to change all your procedures and each reference to the file in your program to make it use a new file, and you want both files to be able to co-exist on the computer.

To achieve my second, third and seventh objectives, I need to create an executable program that I can distribute freely, and I need to make the conversion run right after the installation. You may be tempted to make the conversion run the first time the new version of the program is run, but there are some problems with this approach. For example, you may have various executable files in your application, and you don't know which one the user will run first. You also may have users who install the new version of your program, but don't run it for a long time.

Objectives four and five can be achieved by deleting the data from the old file as you add it to the new file. Then if you restart the program, it will resume converting the data that is still in the old version of the file. If the conversion completes and the program is run again, you can check if there are any records left in the old file, and not run the conversion at all if it's empty.

The sixth objective is easy to achieve. Just back up the database before starting the conversion. If everything fails, you can always put things back the way they were before the conversion started.

The original version

First I'll look at a version 1 of a sample application with the original files. Then I'll go through the steps to create version 2 of the program, in which the files have additional and changed fields and keys.

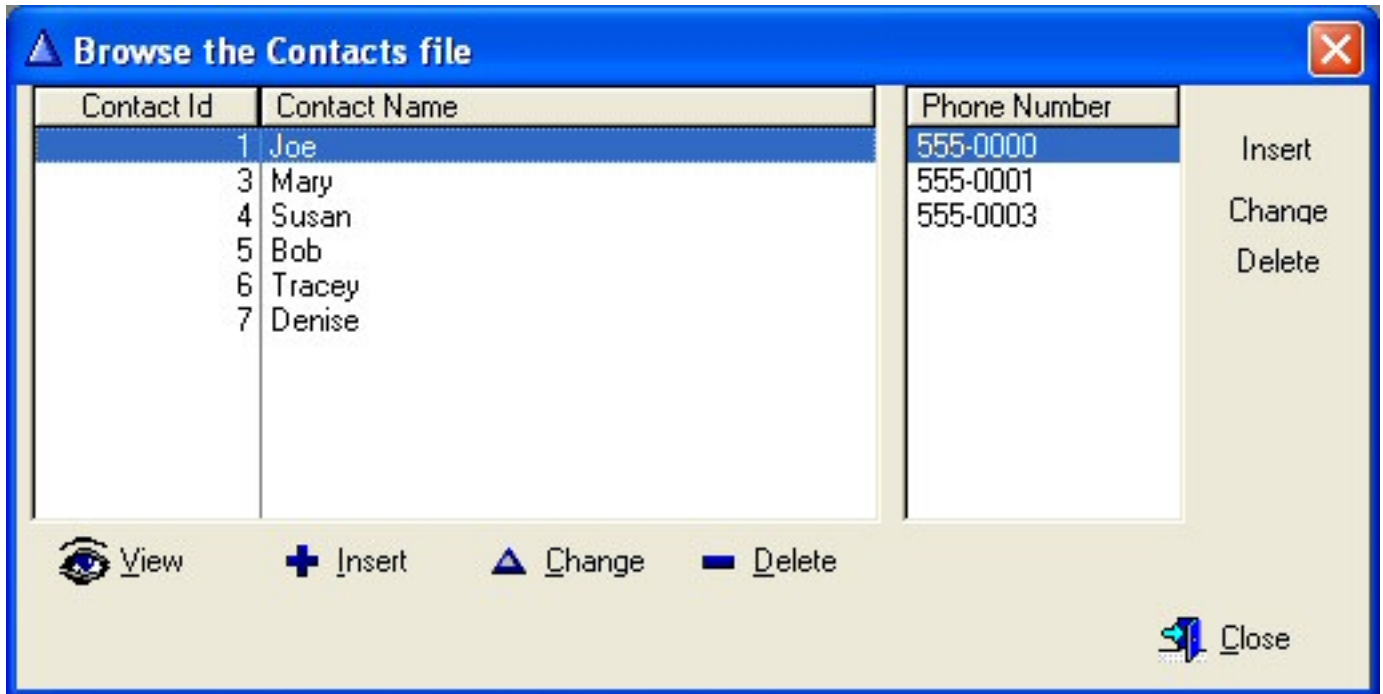


Figure 1. The original application

The application I'm using for demonstration purposes is very simple. Its database consists of two files. The `Contacts` file has a contact id and a contact name. The `Phones` file has a phone id, a contact id and a phone number. There is a one-to-many relationship between the contacts and the phone numbers; in other words, a contact can have many phone numbers. When you click on a contact in the browse box on the left of Figure 1, the browse box on the right displays the phone numbers for the selected contact.

Now let's look at the dictionary for this application.

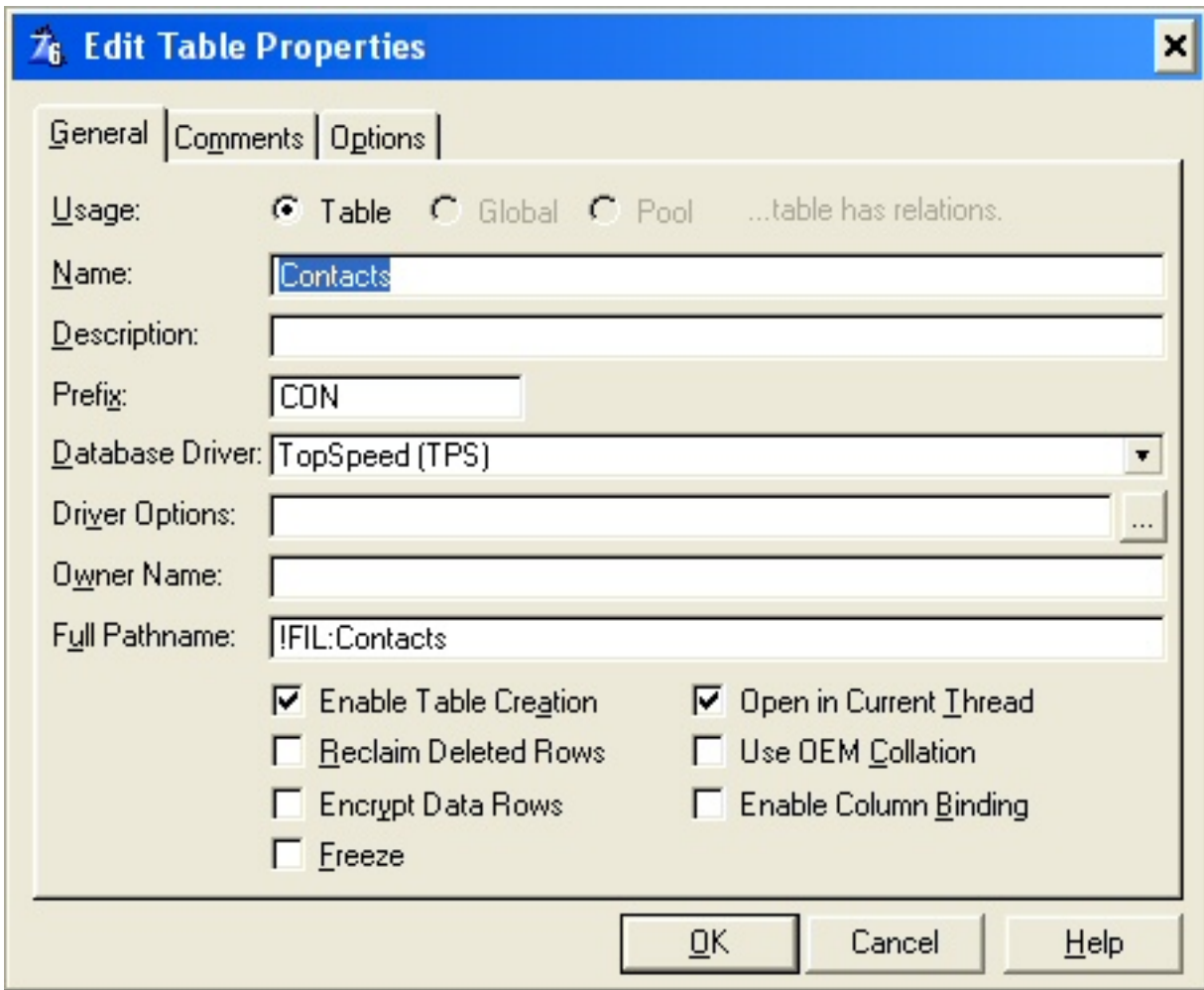


Figure 2. Contacts table properties

Figure 2 shows the property window for the contacts file. Note that I'm using a variable for the Full Pathname . This is not required, but it's usually a good idea. The property screen for the Phones file is pretty much the same. I'm using the `FIL:Phones` variable to store the path and file name. Figure 3 shows where I defined the global file name variables for the two files.

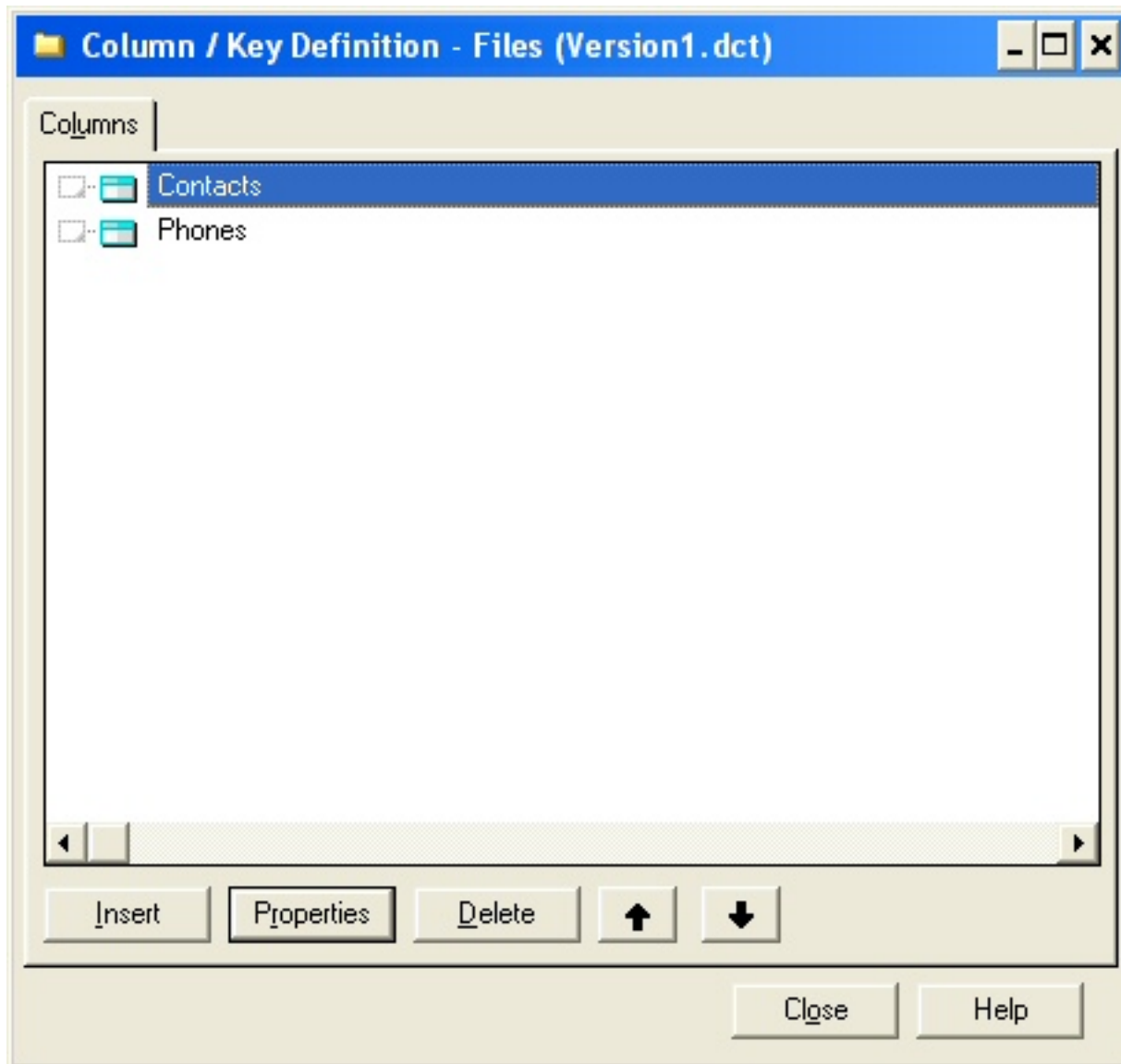


Figure 3. Global file name definitions

I defined two identical sets of global file names (Files and OldF). The only difference between them is the prefix for the variables. Figure 4 shows the entire dictionary.

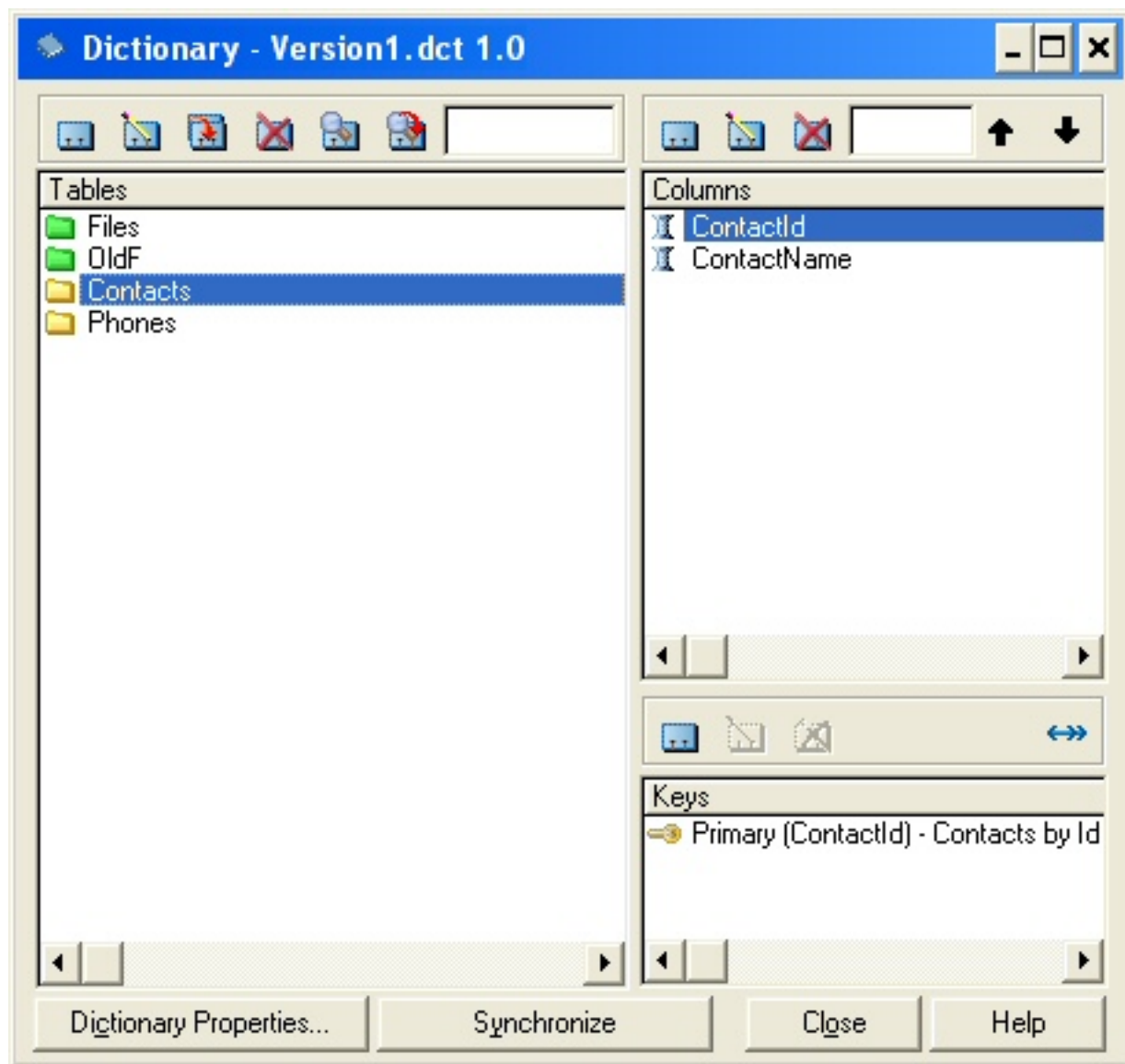


Figure 4. The dictionary with tables and global variables

Coding version 2

Now it's time to start to work on version 2, which will include the following improvements:

1. I want to add a key to the `Contacts` file by name.
2. I want to add an email address column to the `Contacts` table
3. I want to make the phone number field 20 characters wide instead of 10 so I can enter international phone numbers.

4. I want to add a column to the Phones file for entering an extension number

As my starting point for version 2 of the program, I create copies of the application and dictionary file. In the original application these files were called Version1.app and Version1.dct. The new versions are called Version2.app and Version2.dct. The plan is to modify the new versions of the files to create my new application.

The first step is to create copies of my files in the dictionary editor. Just copy the file definitions of the files you want to change (in this case both files) and paste them back in the dictionary with a different name (I prefixed each file name with the word old). Your dictionary should look like the one in Figure 5

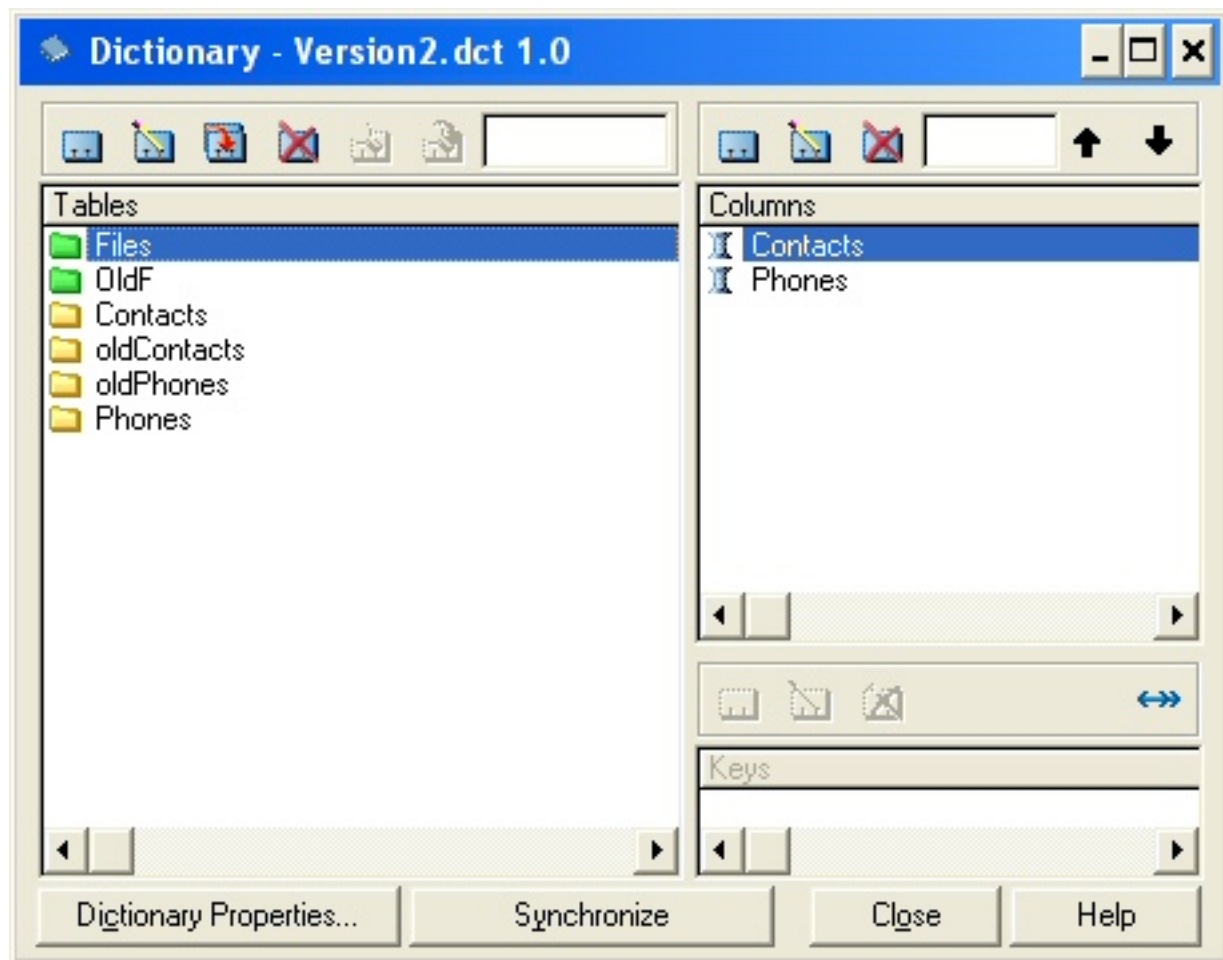


Figure 5. Copying the file definitions

Next I need to change the Full Pathname value in the property window of each of the newly created files, as shown in Figure 6. The OLD: is the prefix associated with the global variables in the OLDF structure in my dictionary.

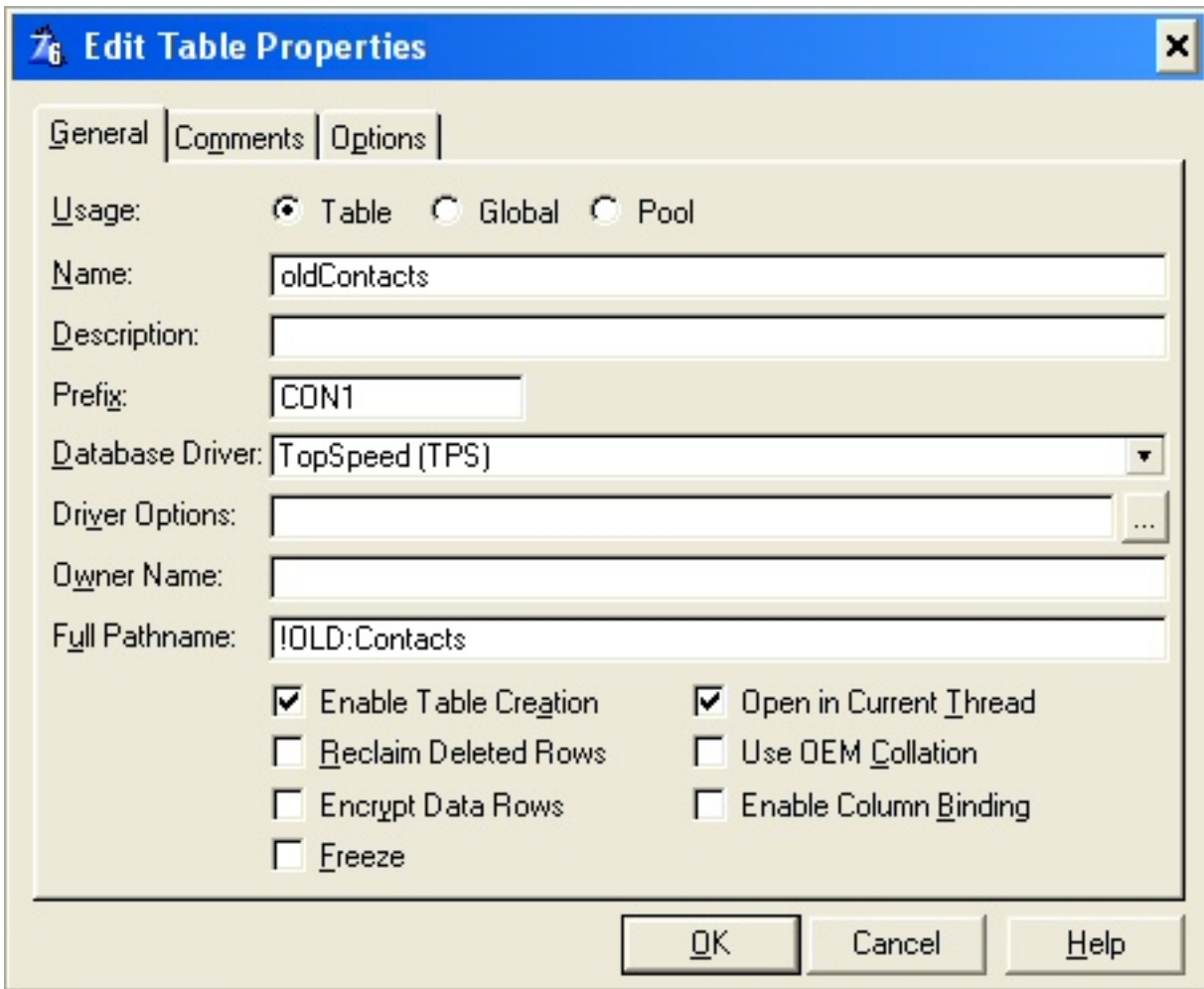


Figure 6. Changing the file pathname

Now that I have saved the present definition of the files, I can make the changes needed for version 2 of my application. I'm not going to describe how to make these changes, since most of you know how to do this. Simply edit the definition of the `Contacts` and `Phones` file in the dictionary editor. If you need help using the dictionary editor, you should consult the Clarion documentation.

After making the dictionary changes, I can make the changes to the application. I can populate the new fields on the windows and change the widths of the controls to match the new column sizes. I also want to change the browse box for the contacts to use the new key by name instead of the primary key by contact id.

Since I used variable names for the paths, I need to change the section of the program where they are initialized. Figures 7 & 8 show how the path variables are initialized in version 1 and version 2 respectively. If you hard-coded the file paths, you do not have to make this change. Simply hard-code a new name for each of the new files. Notice I put the version number in the file name. You can use any naming convention you

prefer for your files, just make sure that you use different names in version 2 and version 1.

```

ThisWindow.Init PROCEDURE

ReturnValue          BYTE,AUTO

! Start of "WindowManager Method Data Section"
! [Priority 5000]

! End of "WindowManager Method Data Section"
CODE
! Start of "WindowManager Method Executable Code Section"
! [Priority 300]

! Enter procedure scope
GlobalErrors.SetProcedureName('Main')
! [Priority 501]
FIL:Contacts = 'Con001.tps'
FIL:Phones = 'Pho001.tps'
    
```

Figure 7. Specifying the file names (version 1)

```

ThisWindow.Init PROCEDURE

ReturnValue          BYTE,AUTO

! Start of "WindowManager Method Data Section"
! [Priority 5000]

! End of "WindowManager Method Data Section"
CODE
! Start of "WindowManager Method Executable Code Section"
! [Priority 300]

! Enter procedure scope
GlobalErrors.SetProcedureName('Main')
! [Priority 501]
OLD:Contacts = 'Con001.tps'
OLD:Phones = 'Pho001.tps'

FIL:Contacts = 'Con002.tps'
FIL:Phones = 'Pho002.tps'
    
```

Figure 8. Specifying the file names (version 2)

At this point I can compile and run my Version 2 application and it works. Since I'm creating new files I don't get error 47, but all the data is gone because it's still in the old files. The next thing I need to do is create a conversion program that will copy the data from the old files to the new ones. I'll describe that process [next week](#).

[Download the source](#)

[Abe Jimenez](#) started programming in the late 70s in RPG on the IBM System 34. Towards the late 80s he began using Clarion 2.1 for DOS. Over the years he has programmed in all versions of Clarion, but not continuously. He now develops and markets an application called TechQwest Office (written in Clarion) for the distribution and wholesale vertical markets. His company develops Window, Web and Mobile Device applications in Clarion, VisualStudio.net and Java.

Reader Comments

[Add a comment](#)

- [» Hi Abe, How to you manage the situation where the user...](#)
- [» Hi Bruce, This, like most magazine articles is only...](#)
- [» Sounds like hard work to me, if your app has 25 tables...](#)
- [» Trevor, Capesoft produces many fine products, including...](#)
- [» Trevor, I know that Capesoft produces many great 3rd...](#)

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

Notifying ClarioNet Clients With PushWindow

by Jason Sylvestre

Published 2005-02-25

[ClarioNet](#) is a great product which lets you get your program working over the Internet quickly, and usually with few code changes needed. However, one problem that I have found is that, as a thin client, the ClarioNet server responds only when the client makes a request, to minimize the amount of network traffic. This can be annoying when the server side needs to complete a batch process and you want to inform the user that everything is okay and the client has not frozen up. ClarioNet provides a way around this with PushWindows. The server "pushes" the window and its contents to the client side.

There are several issues that you should be aware of when using PushWindows. The most important is that if another window, or a `message ()` dialog is opened, it will hang the program. Also, the number of times a PushWindow is opened or updated increases the network traffic and can slow down the process, as all the information has to be sent to the client. The `ClarioNet:UpdatePushWindow(CWindow)` function updates the entire window and not just individual controls. If you have a list box with a lot of information in it, all that will be sent each time you update or open the PushWindow.

I've found PushWindows useful when importing and validating data. The following simple steps show how to create a PushWindow for a batch process.

1. Create your procedure and the display window normally. This isn't required, but I've found it's easier to work with a normal procedure than to create a source procedure just to call PushWindow.
2. Open your procedure in source and find the window structure. Copy

your window structure, paste and rename the window label. I suggest naming it CWindow for consistency. (Duplicate controls *will* cause compiler warnings).

```

! Window Structure
Window          WINDOW('UIF Import'),AT(, ,317,215),FONT('Microsoft Sans Se
                LIST,AT(9,5,234,169),USE(?ResultsList),SCROLL,USCROLL,FO
                LIST,AT(249,5,57,169),USE(?ProuCountList),FONT(, , ,FONT:
                PROGRESS,USE(LGL:OverAllProgress),AT(9,180,298,5),RANGE
                PROGRESS,USE(LGL:FileProgress),AT(9,188,298,5),RANGE(0,
                BUTTON('&Print'),AT(4,200,45,14),USE(?Button1),SKIP,DIS
                BUTTON('Do Nothing'),AT(101,200,45,14),USE(?DoNothing),
                BUTTON('E&xit'),AT(268,200,45,14),USE(?Button2),SKIP,DIS
                END

! [Priority 7500]
CWindow        WINDOW('UIF Import'),AT(, ,317,215),FONT('Microsoft Sans Se
                LIST,AT(9,5,234,169),USE(?ResultsList),SCROLL,USCROLL,FO
                LIST,AT(249,5,57,169),USE(?ProuCountList),FONT(, , ,FONT:
                PROGRESS,USE(LGL:OverAllProgress),AT(9,180,298,5),RANGE
                PROGRESS,USE(LGL:FileProgress),AT(9,188,298,5),RANGE(0,
                !!BUTTON('&Print'),AT(4,200,45,14),USE(?Button1),SKIP,DIS
                BUTTON('Do Nothing'),AT(101,200,45,14),USE(?DoNothing),
                !!BUTTON('E&xit'),AT(268,200,45,14),USE(?Button2),SKIP,DIS
                !!LIST,AT(249,5,57,169),USE(?ProuCountList),FORMAT('21R
                END

```

Figure 1. Duplicating the window structure

3. Progress Bars cannot have their RangeHigh and RangeLow set dynamically. Keep them at 0 and 100 and then do your own calculation: $(\text{CurrentPosition} / \text{MaxPosition}) * 100$.
4. In the procedure's generated source, search for where the normal window is opened. In an embed, before the normal window is opened, put the code to open the PushWindow. Always wrap your PushWindow code with a check to see if ClarioNet is being used to prevent any problems if it is run locally.

```

IF ClarioNetServer:Active()
    ClarioNet:OpenPushWindow(CWindow)
END

```

5. In the OpenWindow Embed (Before the generated code), put your main driver code. To update the display use the following code:

```

IF ClarioNetServer:Active()

```

```

        ClarioNet:UpdatePushWindow(CWindow)
    END
    DISPLAY

```

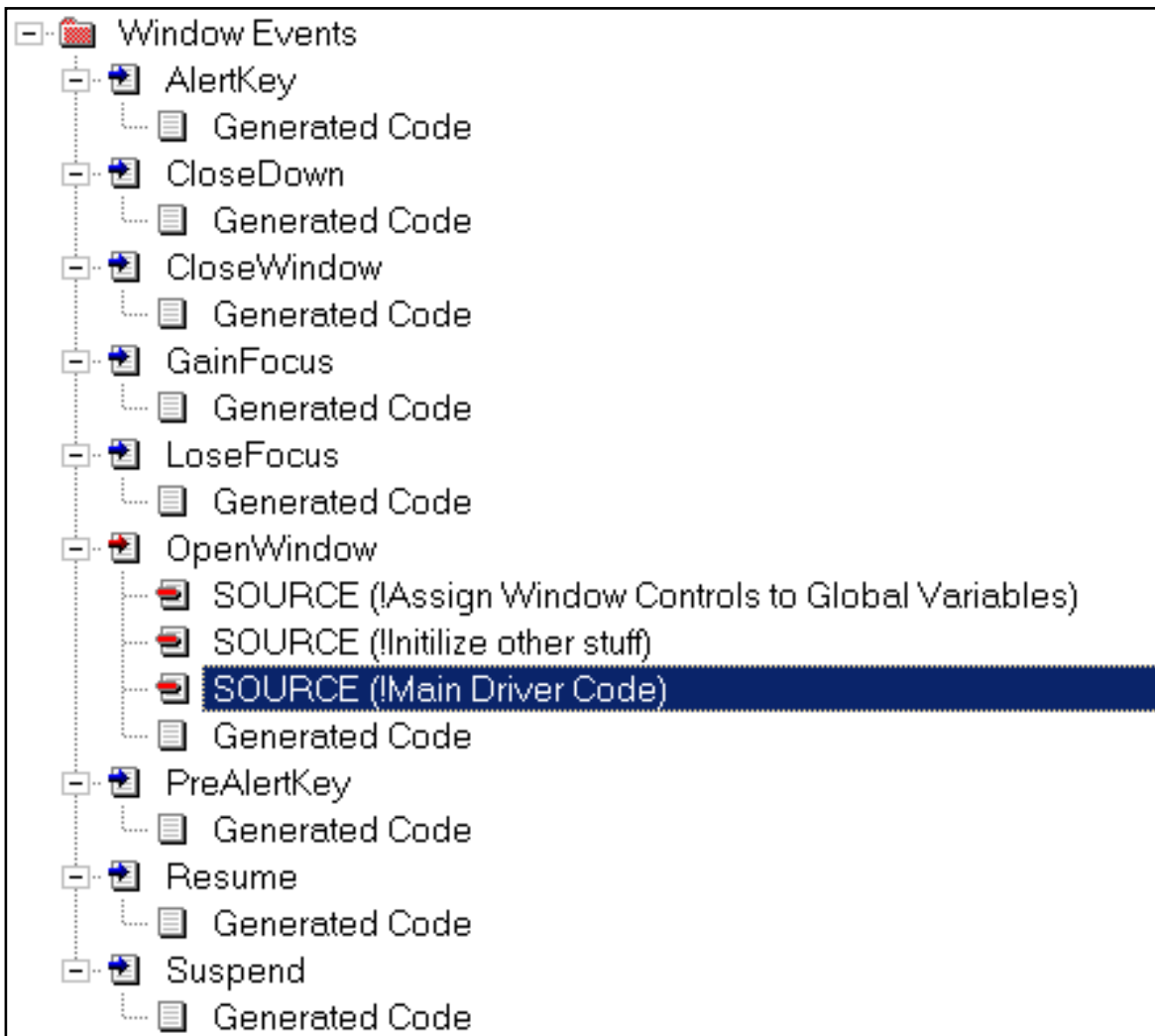


Figure 2. Embedding the source code

- To prevent too many updates, you could use a time-based method, or use the modulus operator to limit the number of updates. Remember, window timers don't work with ClarioNet.

```

IF CurrentPosition % 10 = 0
    IF ClarioNetServer:Active()
        ClarioNet:UpdatePushWindow(CWindow)
    END
END

```

- When you are done processing in the OpenWindow Embed, put the following code:

```
IF ClarioNetServer:Active()  
    ClarioNet:ClosePushWindow(CWindow)  
END
```

8. The other Window will then open and display.

If you plan to update the PushWindow in procedures other than where the PushWindow is declared, you can save the reference to the PushWindow into a variable of type &WINDOW. Simply assign it with &= Cwindow. Just remember that none of the procedures can call a message () or open another window without closing the PushWindow first. I use a global variable for simplicity when I need to do this.

If you need to display a message dialog to the user, you need to close the PushWindow first. After the message has been displayed, you can open the PushWindow again. It might be better to design a log that will be displayed after the process that uses the PushWindow completes.

The ClarioNet manual has more information on PushWindows, but I've found the above way of using PushWindows to be very effective. This technique applies to ClarioNet 1.4, Clarion 5.5 but should work with other versions.

Reader Comments

[Add a comment](#)

Copyright © 1999-2005 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.