Clarion Magazine

Clarion News

- » SV News Server Downtime
- » Simsoft Templates Clarion 6.3 Compatible
- » SysAni, SysList, SysTree for Clarion 6.3
- » dpQuery 2.07
- » Australian DevCon 2006 Update
- » ClarionTools Wizards with Clarion 6.3 Support
- » IP Driver 2.3 Demo
- » EasyResizeAndSplit 2.11
- » 6.3 Compatible CwEventLog, FrameText, MessageEx and RichReport
- » All Data Equity Assistants Now Gold
- » FRB 3.02 released, Clarion 6.3 compatible
- » Huenuleufu Development Moving Sale
- » FullRecord 1.21
- » ClarionTools Wizard Product Support for Clarion 6.3
- » EasyExcel 3.08 for 6.3
- » Huenuleufu Products Are 6.3 Compatible
- » CPCS v6.30 for Clarion 6.3
- » RPM, AFE, PNet for 6.3
- » PD Drops 6301 for Clarion 6.3
- » PD World Wide Address Formats 6301 for Clarion 6.3
- » Gitano Utilities 6.3 Compatible
- » SetupBuilder 5.3 Released
- » Free Clarion Book
- » CapeSoft Clarion 6.3 Compatible Downloads.
- » CapeSoft KeepingTabs

- » CapeSoft BreakIn
- » xClarionSwitcher 1.05
- » iQ-XML 1.13
- » xXPpopup 1.5
- » PrintWindow Build 110
- » Data Equity Major Product Updates
- » Clarion 6.3 Released
- » FullRecord 1.20
- » EasyCOM2INC 2.07 Fix
- » xButton 1.7
- » Clarion Connection DNS Issues
- » Nikasoft Email Server Restored
- » Clarion 6.3 Release This Week
- » Free Pervasive
- » Data Equity Assistants Release 2.00.00.26
- » PrintWindow Build 109
- » Enabling Simplicity Sale Extended To January 12
- » EasyCOM2INC_WMI 1.01
- » ComSoft Updates
- » Aussie DevCon Registrations Now Open
- » CDC Week #17 Winner and Grand Prize Winner
- » Epison Concepts Becomes Epsilon Network, LLC

[More news]

Podcast

[Track lists, more podcasts]

Latest Free Content

[More free articles]

Clarion Sites

- » Lodestar Software (CWaddons)

Clarion Blogs

- » DevDawn

Latest Subscriber Content

App Shutdown Options in Clarion 6.2

Clarion 6.2 introduced two very handy new features; the ability to automatically shut down a Clarion application when Windows shuts down (or the user logs off), and the ability to easily shut down any application with a single function call.

Posted Tuesday, January 31, 2006

Recovering From The Big Crash

Nope, this isn't an article - it's what I'm doing after my development machine suffered a catastrophic hard drive failure over the weekend. If you sent me an email last week and haven't yet had a reply, please send again. But please be patient if I don't reply right away - since I'm having to reinstall everything, it'll be another day or so before things are back to normal around here.

Posted Tuesday, January 31, 2006

First Look: Clarion 6.3

Dave Harms upgrades to Clarion 6.3, and along the way relearns a few lessons about keeping the Clarion development environment under version control.

Posted Friday, January 27, 2006

Anyone Home on the Network?

IsNetworkAlive, IsDestinationReachable, GetLastError, FormatMessage, GetDriveType, WnetGetConnection - the odyssey to know the status of the network began with a printer error. It was the most peculiar printer error Steve Parker and Henry Plotkin had ever seen.
Posted Friday, January 20, 2006

Next Month Anniversary Date Function

Inspired by Carl Barnes Better Date article, John Griffiths offers up this function to get anniversary dates on specific days of the month.
Posted Thursday, January 19, 2006

Edit-In-Place: Getting User Confirmation

No good deed, it is said, goes unpunished. And now Henry Plotkin comes to realize that Edit-in-place is ready, willing and able to punish the good-deed-doing Clarion developer. Perfectly willing.
Posted Friday, January 13, 2006

Completely Dynamic Listbox Formatting?

The Point-of-Sale (POS) application Steve Parker manages allows users to select subsets of inventory for a variety of purposes. These include order recommendations, physical inventory, creating store-wide sales and several other things. What started off as a reporting feature eventually became a browse requirement. And that meant creating not just dynamic filters, but dynamic list box formats.
Posted Wednesday, January 11, 2006

PDF for December 2005

All Clarion Magazine articles for December 2005 in PDF format.
Posted Tuesday, January 10, 2006

A Better DATE Function

Carl Barnes improves on the standard Clarion DATE function with a version that corrects a bug and works with negative month and day values.
Posted Monday, January 09, 2006

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

Printed Books & E-Books

E-Books

E-books are another great way to get the information you want from Clarion Magazine.

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

Printed Books

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

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

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

From The Publisher

About Clarion Magazine

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

Subscriptions

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

Satisfaction Guaranteed

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

Dave Harms

Clarion Magazine

Clarion News

Search the news archive

Evolution Report Export 1.09

Evolution Report Export 1.09 exports to Excel, HTML, ASCII, CSV, Report and XML from the standard Clarion Report Preview. Clarion 5, Clarion 5.5, Clarion 6. ABC or Clarion (Legacy) Templates.
Posted Friday, February 24, 2006

Evolution Edit in Form 1.04

The Evolution Edit in Form 1.04 template transforms your standard Forms into an Edit in Place. You can use the form, as always, and the template resizes the form window on top of the browse row, simulating the Edit in Place behaviour. You can use lookups, dropdowns, calendars, tooltips, etc. and all the form embed points. For Clarion 5.5 and Clarion 6.x. Both ABC and Legacy.
Posted Friday, February 24, 2006

Clarion FreeImage Project 3.8.0.3

The Clarion FreeImage Project 3.8.0.3 is now available for download. This update includes an installer built with SetupBuilder. New examples show how to use the File Dialog and Clipboard classes without the FreeImage class or Image control. The Clarion FreeImage Project is an open source project built around the FreeImage raster processing library.
Posted Friday, February 24, 2006

18,960 Ace Icons For $49.95

You now get 18,960 icons in either the Ace Brights or Ace Lights icon sets, for $49.95 each. In the Roma set you get 7020 icons. Both the Ace and Roma icon sets are being expanded; the Roma set will soon have over 10,000 icons. Anyone who purchases (or has purchased in 2006) the Ace or Roma icon sets will receive the new images for free, when they are released.
Posted Friday, February 24, 2006

EasyExcel 4.00

EasyExcel 4.00 is now available. New methods include: SetPageHeaderFont - sets page header font; SetPageFooterFont - sets page footer font; FreezePanes - freezes the split panes. Changed methods include: GetSelection - added new parameter and method now returns errorcode; List2Excel - method was re-written to support the date and time columns and was fixed a bug of incorrect columns format setting. Changed templates include: PageSetup code template - now includes the ability to change page header/footer font. The upgrades from 1.0 to 2.0/3.0 and from 2.0/3.0 to 4.0 are available at ClarionShop for only US$39. Registered users that purchased version 2.0/3.0 after September 1st, 2005, will receive this upgrade free of charge.
Posted Friday, February 24, 2006

xPowerManager 1.02 Freeware

xPowerManager 1.02 is a free tool to automatically shutdown your PC on a timer. You can use a countdown timer or an alarm clock timer.
Posted Friday, February 24, 2006

xNotes 1.7

xNotes 1.7 is now available, and is compatible with PowerXP-Theme 2. xNotes supports SingleExe, MultiDll (Local Mode, Standalone Mode), 32-bit. Compatible with Clarion 6.x (6.3, 6.2, 6.1), Clarion 5.5 and Clarion 5.
Posted Friday, February 24, 2006

xAppWallpaper Template 1.9

xAppWallpaper Template 1.9 is now available. New in this version: Modification in Template for setting xAppWallpaper items in frame menu; You can set/change font, color, heght, width, offset etc; Cmpatible with Clarion 6.x (6.3, 6.2, 6.1); New installation kits

prepared with SetupBuilder 5.3; pdated Demonstration programs and Installation Kit for Clarion 6.x, Clarion 5.5 and Clarion 5. xAppWallpaper Template is $19.
Posted Friday, February 24, 2006

SetupBuilder 5.3 Build 1414

SetupBuilder 5.3 Build 1414 is now available. As always you may use the "Check for Updates" option within SetupBuilder to auto-update your current SB5 version.
Posted Friday, February 24, 2006

Insight Graphing Review

A short review of Capesoft's Insight Graphing, by Ennis Baggott, is now available on DeveloperReviewed.com.
Posted Friday, February 24, 2006

xAccess Manager Lite 2.1

New in xAccess Manager Lite 2.1: xAccessManager is now compatible with PowerXP-Theme 2. xAccess Manager Lite is a class and templates that adds user access rights and privileges to your program.
Posted Friday, February 24, 2006

RTFCreator Freeware

Arnor Baldvinson has a new free tool called RTFCreator, which was created out of the need to fill in RTF fields for display purposes only. RTFCreator lets you create an RTF text and display it as formatted RTF, RTF code and Clarion-ready RTF code. Clarion-ready means that all the { have been doubled up to {{ etc. and it is ready to be pasted into a clarion string variable. The simple RTF editor that comes with RTFCreator allows you to save and open RTF files. The zip file is about 1.6Mb download and the exe is compiled local in Clarion 6.3, so you can run this without problems no matter what version of Clarion you are working with.
Posted Friday, February 24, 2006

iQ-XML Now Faster

Robert Paresi made a small change with the help of John Christ, and now the Parser portion of iQ-XML is 300 (or more) times faster. For example: Loading one file took 2.48

seconds versus the change in V1.15 which now takes .04 seconds. (The change is only in C6.3 but will be applied to the 6.2 version).
Posted Friday, February 24, 2006

Dictionary Assistant 2.10

Dictionary Assistant 2.10 is now available. This release is free of charge to all registered customers of Dictionary Assistant 2. Enhancements in this release include new reporting options as well as enhanced exporting capabilities. DA 2.10 now offers tighter integration with Visual Studio via the broadened functionality of the XSD Schema Export Utility. As always you may use the "Check for Updates" option within DA or download the update directly from the web site.
Posted Friday, February 24, 2006

ImageEx 3.5

Version 3.5 of ImageEx has been released. Consider this an interim release between version 3 (which is one and a half years old now) and the planned version 4 (for which there is no timeline as yet). This version contains much of the core code initially intended to go into version 4. New features include: A new class (ImageExTwainClass) for dealing with twain devices - including multi-page image acquisition; Another new class (ImageExPrinterClass) for printing images directly to the printer, also for multi-page print jobs: combined with the new twain class, you could create a "copy" function with just a few lines of code; A complete rewrite of the thumbnailer class which now heavily overloads the CreateThumbnail method to provide more intuitive usage than before; Added functionality to the PdfSaver class to allow the creation of multi-page PDF files (one image per page); Added the ability of loading images from / saving images to memory (a block of memory identified by a HGLOBAL handle); Added the ability of loading images directly from the web with standard URL notation: (http|ftp)://[username[:password]@]hostname[:port]/path. Numerous other improvements and bug fixes. ImageEx now also supports version 6.3 of Clarion natively, without the "lib" link mode workaround. Also contains native compiles in version 5.0, 5.5, 6.0, 6.1 AND 6.2 plus updated example apps. The upgrade from 3.0 is available at ClarionShop for only US$49 and the one from 2.x costs US$99. Registered users that purchased version 3.0 after August 1st, 2005, will receive this upgrade free of charge. The price of the full version now is US$299.
Posted Friday, February 24, 2006

900 d-Icons for $10

Dave Beggs has added round and octagonal icons to the d-icons set. The package now includes 60 styles in 5 colors and 3 shapes (900 icons) plus the same icons in 256x256 Vista-compatible PNGs and GIFs (1800 files in all) plus blank background icons in case you wish to add to the set yourself. Price is US$19.95. Clarion developers are welcome to a 50% discount by using the coupon CPN0402538605.
Posted Friday, February 24, 2006

iQ-XML 1.14 For C6.3

In iQ-XML 1.14 both C6.2 and C6.3 users have the same functions and enhancements. If you are using the XML:PRIMARYKEYCASCADE() function anywhere in your code, you must revisit the code and make the necessary changes. Failure to do so will cause problems in your code. Use Clarion's "File / Search Files" for "XML:PrimaryKeyCascade" for your code to find any places that need to be changed. Other changes include: Changed the XML:PrimaryKeyCascade() function to handle three different levels of cascading; Added Field name to clear to XML:PrimaryKeyClear() function; Added Settings button to the DebugQueue() screen to show all settings that have been set within the IQXML library; Fixed conversion of and for LoadQueue; Changed code will speed up the Writer functions; IQXML:FORCEFLUSH has been renamed to IQXML:FORCEWRITE simply to make more sense; Minor enhancements to the LoadQueue() function to make it faster when dealing with PrimaryKey and Attribute fields; Various other changes and enhancements.
Posted Friday, February 24, 2006

iQ-XML 1.14b

iQ-XML 1.14b adds the ability to set a ?Progress control target on the XML:SetProgressWindow function so a progress bar on your local window will be updated rather than seeing a child progress window.
Posted Friday, February 24, 2006

xToolTipPro 1.1

xToolTipPro 1.1 is now available. New in this version:Changes in class and template for compatibility with xDataBackup Managers (Pro and Lite) and xAccessManager Lite; Small bugfix in local extension template
Posted Friday, February 24, 2006

xDataBackup Manager Pro 3.03

xDataBackup Manager Pro 3.03 is now available. This release includes some important template changes, and further compatibility with PowerXP-Theme 2.
Posted Friday, February 24, 2006

ClarioNET Source Code Offered

Michael Brooks is accepting offers of interest for source code licenses for ClarioNET, as well as offers of interest in taking over maintenance of the source. Michael originally wrote ClarioNET (which is all Clarion code, except for the compression library) to web-enable Clarion applications for his own structural engineering software business. He is now too busy with long-standing business to continue to support ClarioNET as extensively as needed. Source code licenses are available from $395, but a minimum number of orders must be received. If anyone or any group has interest in taking over ClarioNET entirely please call Michael at 949-645-0441.
Posted Friday, February 24, 2006

DCT2SQL Updated

The DCT2SQL template collection has been updated. DCT2SQL allows conversion of Clarion dictionaries to several SQL backend DDL creation scripts. There are also several utility templates making the conversion from ISAM files to SQL files easier.
Posted Friday, February 24, 2006

xDataBackup Manager Pro 3.02

xDataBackup Manager Pro 3.02 is now available. This is a bugfix release, plus some further changes to support xToolTip and xToolTipPro.
Posted Friday, February 24, 2006

StrategyOnline Takes Over Plugware Products

Gary James, who recently left CapeSoft to focus on his own company, StrategyOnline, has taken over the development of various products written by Plugware Solutions (Andy Ireland). Gary has worked closely with Andy over the past couple of years on products like File Explorer, Office Inside and J-Flow, and am very excited (and honored) to be taking over some of his code. The first Plugware product released by StrategyOnline is PlugIT, the memory leak checker and memory allocation engine that Andy wrote
Posted Friday, February 24, 2006

FinalStep 2.01

New for FinalStep 2.01: Export and Import global settings for easy use in multi-DLL apps; Now compatible with the ERS template, Power XP Toolbar template and Power XP Theme template; Photos files renamed to English, to avoid the use of extended characters that can prevent the instalation of FinalStep on unicode devices. FinalStep allows you to change completely the look of your program in a few minutes.
Posted Monday, February 06, 2006

RADIntelliSense Website Moved

Kelly Major points out that the RADIntellisense web site has moved to a new location.
Posted Monday, February 06, 2006

D-Icons

Earlier this year Dave Beggs was doing some contract programming for a large multi-national company. Large companies have lawyers. Lawyers don't always know much about the field in which they give advice. Anyway, the lawyers concerned here wanted Dave to sign a document saying that all the artwork included in my programming was his own work. So he had to buy an icon editor and make icons. He's selling them cheaply in the hope they help someone else. The package includes 60 styles in 5 colors (300 icons) plus the same icons in 256x256 vista compatible PNGs and GIFs (900 files in all), plus blank background icons in case you wish to add to the set yourself. Dave is selling them for $US19.95. Clarion developers are welcome to a 50% discount by using the coupon CPN0402538605.
Posted Monday, February 06, 2006

xDataBackup Manager Pro 3.01

New in xDataBackup Manager Pro 3.0: Rewritten for all source code, no DLL; Uses standard TRN for localization; Modification in template for setting xDataBackup Manager items in frame menu; You can assign HotKeys for xDataBackup Manager items in frame menu; xDataBackup Manager Pro now compatible with PowerXP-Theme 2; New set of icons used in window of xDataBackup Manager Pro; Some small bugs were fixed; Compatible with Clarion 6.x (6.3, 6.2, 6.1) and Clarion 5.5; New installation kits prepared with SetupBuilder 5.3.
Posted Monday, February 06, 2006

xTipHotKey 3.0

New in xTipHotKey 3.0: Minor bugfix to template for correct joint operation with xXPframe, xNotes, xDataBackupManagers, xAccessManager, xTipOfDay.
Posted Monday, February 06, 2006

Clarion Handy Tools Build 10A1.0

A new CHT build is available: 10A1.0. Use your WEBUPDATER tool as usual to bring your tool kit up to date.
Posted Monday, February 06, 2006

Clarion Magazine

App Shutdown Options in Clarion 6.2

by Dave Harms

Published 2006-01-31

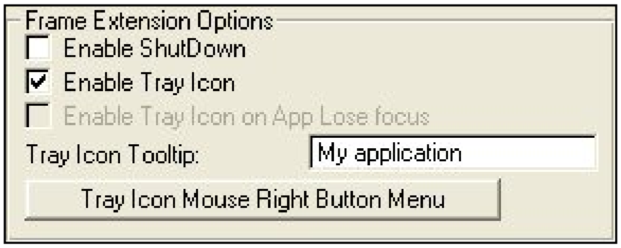
Clarion 6.2 introduced two very handy new features: the ability to automatically shut down a Clarion application when Windows shuts down (or the user logs off), and the ability to easily shut down any application with a single function call (I know, you’ve probably encountered this second feature in your own code, but I am *not* talking about a GPF here).

I’ve been waiting a long time for the first of these features. I realize there have been third party solutions for automatic shutdown available for almost an equally long time, but this is something I’ve always felt should be available out of the box. And the function call to trigger a shutdown can be particularly handy for things like closing an application after a period of inactivity.

Using the templates

Both of these shutdown options are implemented as easy-to-use templates. One is a global extension, and the other is a frame extension.

First, the auto-shutdown template. Bring up the application frame (that is, the main menu) procedure’s properties window, and click on Extensions. Choose Class ABC, Frame Extension. Figure 1 shows the template options.



Frame Extension Options

□ Enable ShutDown

☑ Enable Tray Icon

□ Enable Tray Icon on App Lose focus

Tray Icon Tooltip: My application

Figure 1. Frame Extension options

Enabling the shutdown option adds this line to the application frame’s module map:

```
    INCLUDE('WinExt.INC'),ONCE
```

If not already present, a derived `TakeEvent` method is added to the `ThisWindow` (instance of `WindowManager`) declaration:

```
    TakeEvent PROCEDURE(),BYTE,PROC,DERIVED
```

A new class is declared to handle the shutdown notification:

```
    FrameExtension       CLASS(WindowExtenderClass)
    TrayIconMouseLeft2     PROCEDURE(),DERIVED
    TrayIconMouseRight     PROCEDURE(),DERIVED
                         END
```

The `FrameExtension` class is initialized in `ThisWindow.Init`:

```
    FrameExtension.Init(AppFrame,1,0,0{PROP:Icon},'')
```

One of the things that happens in the Init method is a new subclassing procedure is installed for the application frame:

```
    SELF.OrigWndProc = TheMainWindow{PROP:WndProc}
    TheMainWindow{PROP:WndProc} = ADDRESS(WinExtSubClassFunc)
```

Subclassing in this context isn't like OOP subclassing; it simply means that you're installing your own callback procedure so it can intercept the events the Windows GUI is sending to this window. The subclassing function is prototyped at the top of winext.clw, and you can find the source at the bottom of winext.clw. In particular, `WinExtSubClassFunc` responds to two Windows messages, `WM_QUERYENDSESSION` and `WM_ENDSESSION`.

```
    OF WM_QUERYENDSESSION
      IF GloCurrentWinExt.AllowShutDown
         GloCurrentWinExt.OnShutDown=GloCurrentWinExt.ProcessShutDown()
         IF GloCurrentWinExt.OnShutDown
            RETURN(GloCurrentWinExt.OnShutDown)
         ELSE
            RETURN(False)
         END
      ELSE
         RETURN(False)
      END
    OF WM_ENDSESSION
      IF GloCurrentWinExt.AllowShutDown
         NOTIFY(Event:CloseDown,THREAD())
         RETURN(True)
      ELSE
         RETURN(False)
      END
```

When you tell Windows you want to shut down, it sends all applications a `WM_QUERYENDSESSION` message. If any application objects, you'll see the familiar "Application still active" error message. If all applications respond in the affirmative, Windows next sends the `WM_ENDSESSION` message, telling the apps to shut down.

If you've configured the template to allow auto-shutdown, `WinExtSubClassFunc` will post the `Event:CloseDown` message to the current thread, which tells your application to close all windows and terminate.

When each window receives the `CloseDown` event (or a `CloseWindow` event), the `WindowManager.TakeCloseEvent` method executes. If you're in the middle of changing a record, and you try to close that form or the application itself, you'll get an "Are you sure you want to cancel?" message. But if Windows wants to close your app, it just closes, as if you had cancelled the changes. And for the longest time this was the standard answer I got from the Clarion developers of the day – you don't want automatic shutdown of your application, because the user might be in the middle of some important changes. Well, maybe yes, maybe no. Now you can decide.

The system tray

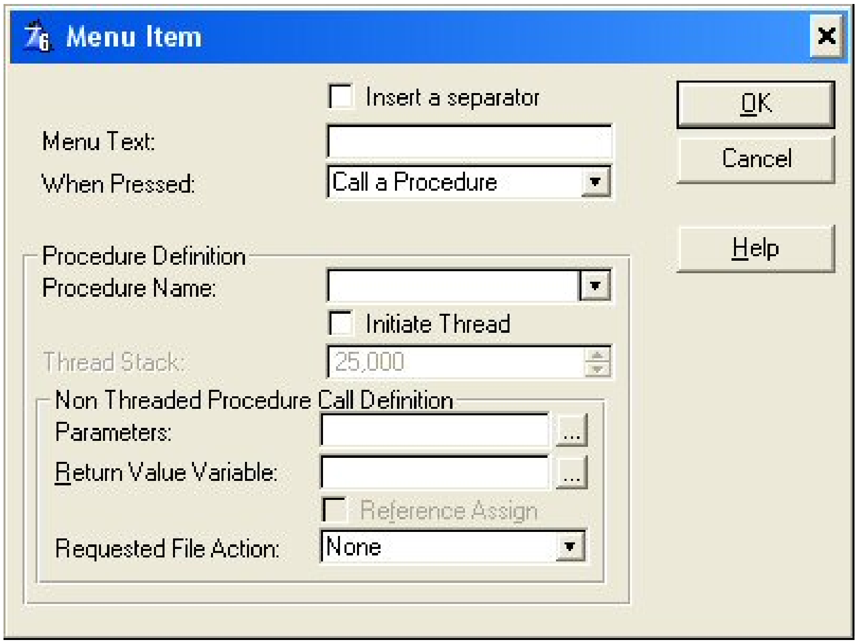
You'll have noticed in the above source listings that the `WindowExtenderClass` also handles system tray (a.k.a. taskbar notification area) operations. In fact, system tray code makes up the better part of the class.

If you've enabled the Enable Tray Icon checkbox in the template, all that changes is the call to `FrameExtension.Init`. The second parameter, `AllowTrayIcon`, is set to 1, and if you've specified a tooltip, that's also passed. Now when you minimize the application, `WindowExtenderClass` (the instance name being `FrameExtension`) uses the `Shell_NotifyIcon` API call to place an icon in the system tray, and then hide the app frame and inactivates itself. Similarly, when you double-click on the icon, `WindowExtenderClass` calls `Shell_NotifyIcon` to remove the icon, and then unhides and activates the app frame.

One nifty thing about the template is it lets you easily add menu options to your system tray icon. Click on the Tray Icon Mouse Right Button Menu button. For each menu item you add, you can:

- Execute a routine
- Call a procedure
- Post event to a control (on the window associated with the extension)
- Run a program
- Do nothing

These options use the expected prompts. Figure 2 shows the options for Call a Procedure.



Menu Item

□ Insert a separator

Menu Text:

When Pressed: Call a Procedure

Procedure Definition

Procedure Name:

□ Initiate Thread

Thread Stack: 25,000

Non Threaded Procedure Call Definition

Parameters:

Return Value Variable:

□ Reference Assign

Requested File Action: None

Figure 2. Calling a procedure from the system tray

One final point about this template – if you've added the extension to the application frame, the option Enable Tray Icon on App Lose Focus will be grayed out. As noted in the help:

> This option is only active if no MDI child windows are opened. This "lose focus" option is designed to be used in applications with only one window (like the IPDRV service manager).

In fact, the template code is looking for the string `'APPLICATION'` in the window structure. So it's not enough that you don't have child windows open; you can't use an application frame.

Clean closedown

The CleanCloseDown template, which is not in any way dependent on the Frame Extension template, makes it easy to initiate an application shutdown from anywhere in your code.

To use the template, go to Global Options, Extensions, and add the CleanCloseDownGlobal template. There is only one prompt on this template; a checkbox that lets you disable the functionality, ostensibly for compatibility testing with third party products.

The template adds a function prototype to the global map:

```
CleanCloseDown()
```

There are also three new global definitions:

```
GLO:CleanCloseDown           BYTE(0)
GLO:CleanCloseDownMainThread LONG
NOTIFY:CloseDown             EQUATE(EVENT:CloseDown)
```

And at the end of the main source module, the CleanCloseDown implementation:

```
CleanCloseDown  PROCEDURE()
 CODE
 GLO:CleanCloseDown = True
 NOTIFY(NOTIFY:CloseDown,GLO:CleanCloseDownMainThread)
```

The main procedure intializes the main thread number:

```
GLO:CleanCloseDownMainThread = THREAD()
```

There is also a new (or modified) `WindowManager.TakeNotify` method in the application frame:

```
ThisWindow.TakeNotify PROCEDURE(UNSIGNED NotifyCode,SIGNED Thread,LONG Parameter)

ReturnValue          BYTE,AUTO

  CODE
  IF NotifyCode = NOTIFY:CloseDown
     POST(EVENT:CloseDown)
  END
  ReturnValue = PARENT.TakeNotify(NotifyCode,Thread,Parameter)
  RETURN ReturnValue
```

You can now call `CleanCloseDown()` from anywhere in your application (say on a timer event, after a period of inactivity). `CleanCloseDown()` posts an event, using the thread-safe NOTIFY mechanism, to the main application thread, which then posts an `EVENT:CloseDown` to shut down the application. As the help notes, in a multi-DLL project you'll need to add this template to any DLLs from which you want to call `CleanCloseDown()`.

Summary

The FrameExtension and CleanCloseDownGlobal templates let you easily add a variety of shutdown and system tray options to Clarion applications. I'm particularly happy to have an out-of-the-box solution to having my Clarion apps shut down automatically when Windows shuts down.

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David produces the Planet Clarion podcast, which he co-hosts with Andrew Guidroz II.

Reader Comments

Add a comment

- » Hi, David. Why can't we add the icon tray extension to a...

Clarion Magazine

First Look: Clarion 6.3

by Dave Harms

Published 2006-01-27

Upgrading software should be, and usually is, painless. The Clarion 6.3 patch itself fell into that category for me (well, except that I had to wait for a new link from SV after the original link went south). The problem is that before getting to the 6.3 goodies, there was a bunch of stuff I wanted/had to do first.

Before installing the upgrade, I realized I was still on C6.2 build 9045, and I had a few patches to apply to get current. And before I did that, I thought it was time I updated my version control repository, so I would have an up to date reference version before proceeding with the patches.

I keep Clarion 6 under version control using Subversion (the heir apparent to CVS) and TortoiseSVN. If you're keeping your apps under version control, you really should keep your development system (including all third party products) under version control as well. If for any reason you have to go back to an earlier version of an application, the ability to restore the exact same development environment could prove invaluable. Sometimes the benefit is less dramatic; the other day I inadvertently opened a second copy of C6 and trashed the registry. It was the work of a moment to restore the registry from the repository. Of course, I could also have followed Danie de Beer's advice in "How To Stop Trashing The Template Registry", and then I wouldn't have had to do the restore at all.

Trashing things seems to be a bit of a habit with me lately.

Anyway, while the fit was on me, I thought I'd update my version control system as well. From the TortoiseSVN About box, I found I was still on 1.2.6, and Subversion 1.2.3. The

Subversion site reported that 1.3 was now available, and the TortoiseSVN also reported a 1.3 version, built against SVN 1.3. Time to upgrade.

I started with Subversion; make sure you get the Windows binaries, unless you like building from source. I installed the upgrade over top of my existing version, and did the same with TortoiseSVN; of the two, only the latter install requires a reboot.

After rebooting, I tried to update the repository with my current Clarion (9045) build. I was able to Add the new files to the repository, although technically that just puts them in the list of files to be added. When I executed the Update, which actually puts the files into the repository, TortoiseSVN just sat there, showing an empty list box, and did nothing.

Well, maybe I made a mistake in installing over top of existing versions. I uninstalled, ran a registry cleaner for that belt and suspenders feeling, and tried again.

Same problem.

Rather than try to track down exactly why TortoiseSVN didn't like either the repository or the current files, and hoping it wasn't a bug in SVN or TortoiseSVN, I made a backup of my C6 directory tree, then deleted all the .svn files from the original. I created a new repository just for C6 (actually keeping project-specific repositories isn't such a bad idea, as repositories do occasionally become corrupt). I imported the files into the repository (TIP: browse to the new repository trunk directory before going to the C6 directory for import). I then created a new C6 directory and checked out the C6 files.

This all took a while, in part because of all those backup files. I suppose I could delete those, seeing as how I have C6 under version control. But I won't.

I applied patch 9046, and that went smoothly. But 9047 failed on an invalid DLL in the BIN directory. After a minute it dawned on me that I'd probably applied an interim 9046 patch, instead of the final 9046 patch. So I reverted to 9045, downloaded the 9046 patch again (which had a different date and size than my previous 9046 patch, confirming my theory).

After that applying the patches went smoothly. I also did a commit and a new tag after each patch, and everything worked fine. I still don't know why commits wouldn't work on the previous repository. This time I immediately made a backup copy of the C6 repository after I finished applying the last patch.

You're probably wondering what all this has to do with the changes in Clarion 6.3. I'll get to that shortly.

What's changed

So what is new in 6.3? First, here's a summary of the official list of changes/new features:

- Server-side SQL auto-incrementing
- MARS on SQL Server 2005
- SQL reconnection improvements
- SQL `ZERODATE` driver string
- SQL case *insensitive* searches are now the default (can be turned off)
- Better driver error info
- `ENCRYPT` will disable logging for drivers that don't otherwise support `ENCRYPT`
- Memory driver improvements
- Dynamic file driver improvements
- Control over driver-wide trace settings
- True angled text strings
- "C6.3 code that allows stored procedures that are stored in packages or referenced by synonyms that return result sets to be correctly called. This code is currently disabled via the conditional compile _VER_C63" Okay, what's that when it's at home?
- Image mirroring
- `TRY`, `CATCH`, `THROW` and `FINALLY` have become reserved words. This is all needed for exception handling in C7/Clarion.NET. Interestingly, the following code compiles in 6.3:

```
code
try
    message('Hey, I''m trying!')
catch
end
```

But the compiler doesn't recognize `THROW`.

- `BufferedPairsClass` can be case sensitive
- `SortHeaderClassType` fully documented, adds inverse order support
- Support for apps to be minimized to the system tray on losing focus (assuming no MDI in use, so more for things like services written in Clarion

than for run of the mill apps)
- New resizing schemes: Centered, and Use Anchor. This looks very interesting.
- EIP improvements
- Version Resource extension template can now be used with the IP Server DLL.

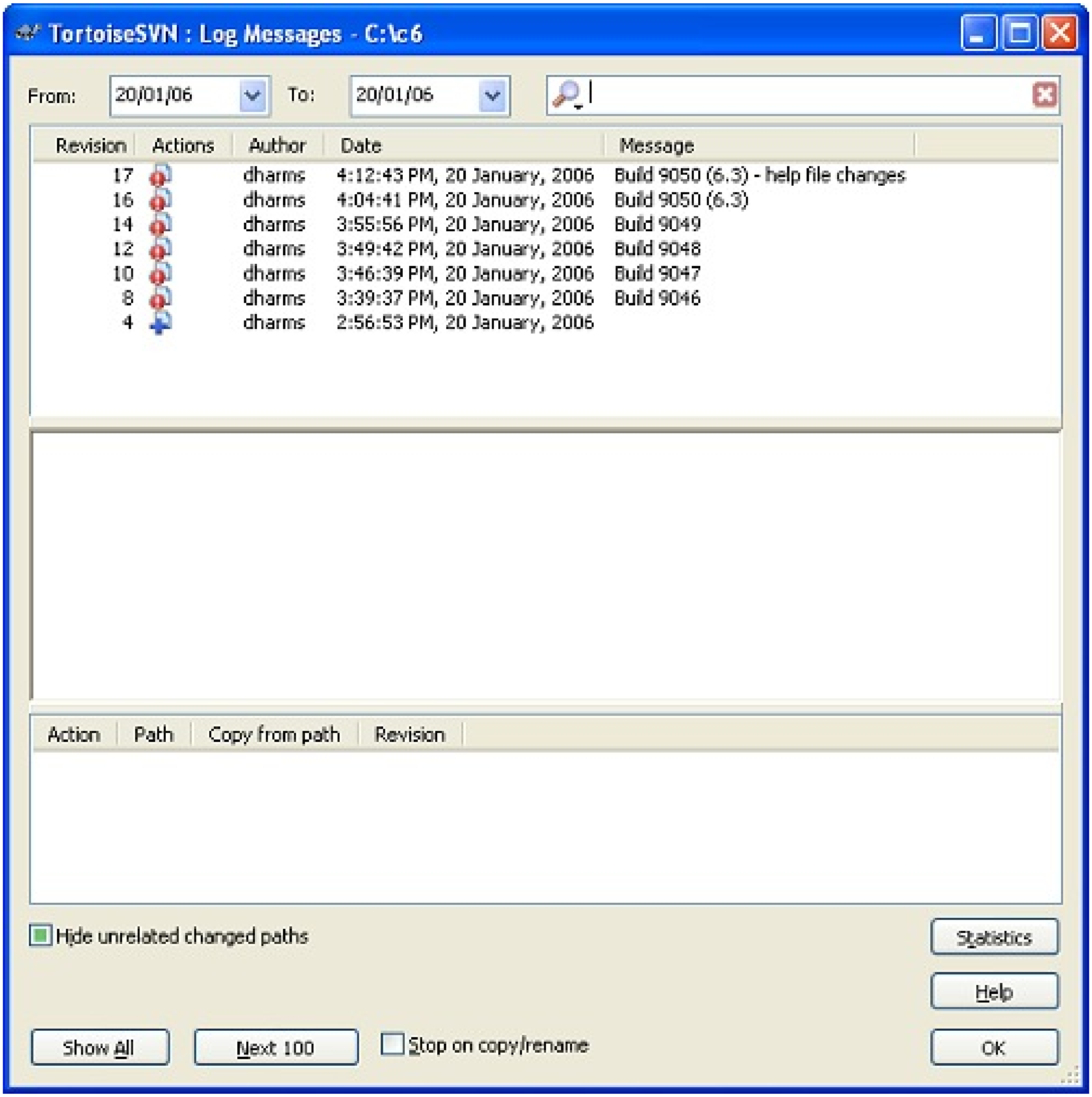
There are also a number of SV blog posts on the 6.3, including:

- Server Side Auto incrementing in Clarion 6.3
- Clarion 6.3 and Multiple Active ResultSets (MARS) on SQL Server 2005 (If you think MARS is a cool SQL Server 2005 feature, just wait until you can write stored procedures and other server-side code in Clarion.NET…)
- Other driver improvements
- New help topic: Clarion Functions used in SQL-based Filter statements, so you can figure out how your filters are being translated into SQL
- Resizing and other ABC improvements

Examining the source changes

You may also find it interesting to see the actual changes made to the shipping source and template files. In the past I've used Beyond Compare for this task; BC is a great tool, and I like it a lot. But TortoiseSVN also has a very nice compare tool. It just isn't very obvious how you use it.

Assuming you have Clarion 6 under version control, as I described earlier, and you have a prior version in the repository, go to the Clarion 6 directory and right-click. Choose TortoiseSVN|Show Log. Figure 1 shows my log view.



From: 20/01/06 To: 20/01/06

Revision	Actions	Author	Date	Message
17		dharms	4:12:43 PM, 20 January, 2006	Build 9050 (6.3) - help file changes
16		dharms	4:04:41 PM, 20 January, 2006	Build 9050 (6.3)
14		dharms	3:55:56 PM, 20 January, 2006	Build 9049
12		dharms	3:49:42 PM, 20 January, 2006	Build 9048
10		dharms	3:46:39 PM, 20 January, 2006	Build 9047
8		dharms	3:39:37 PM, 20 January, 2006	Build 9046
4		dharms	2:56:53 PM, 20 January, 2006	

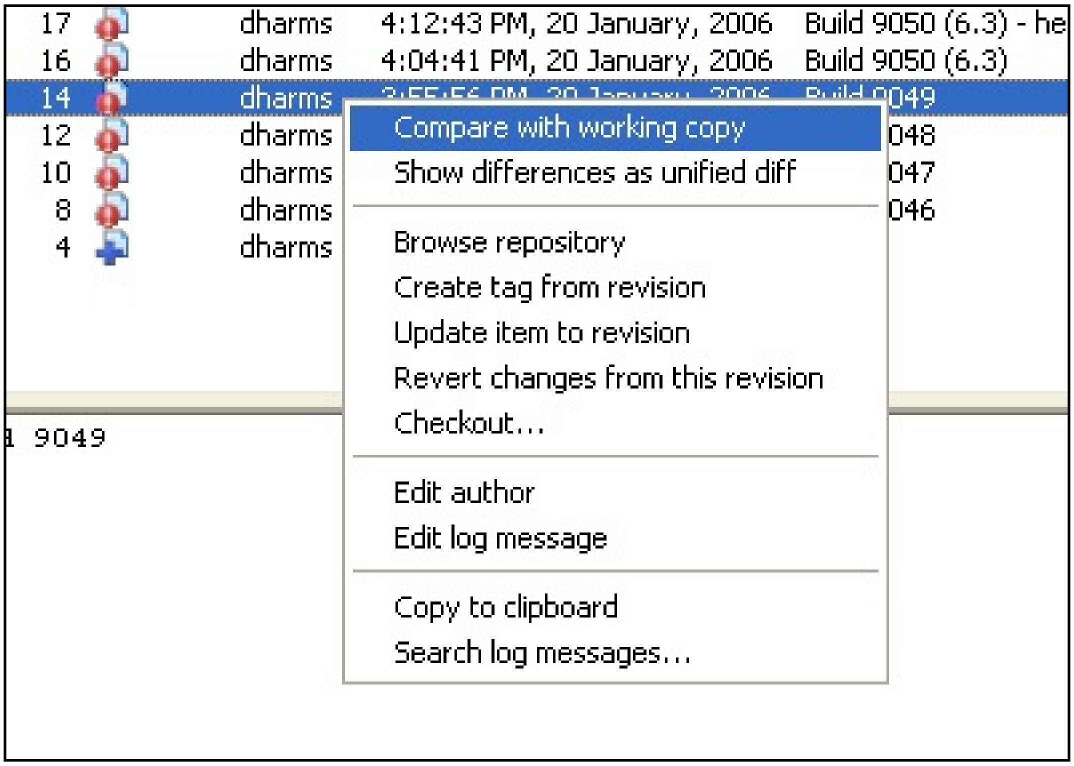
Action	Path	Copy from path	Revision

☑ Hide unrelated changed paths

□ Stop on copy/rename

Figure 1. TortoiseSVN log messages (view full size image)

Right-click on the version you want to compare with the current version, and you will see the context menu in Figure 2.

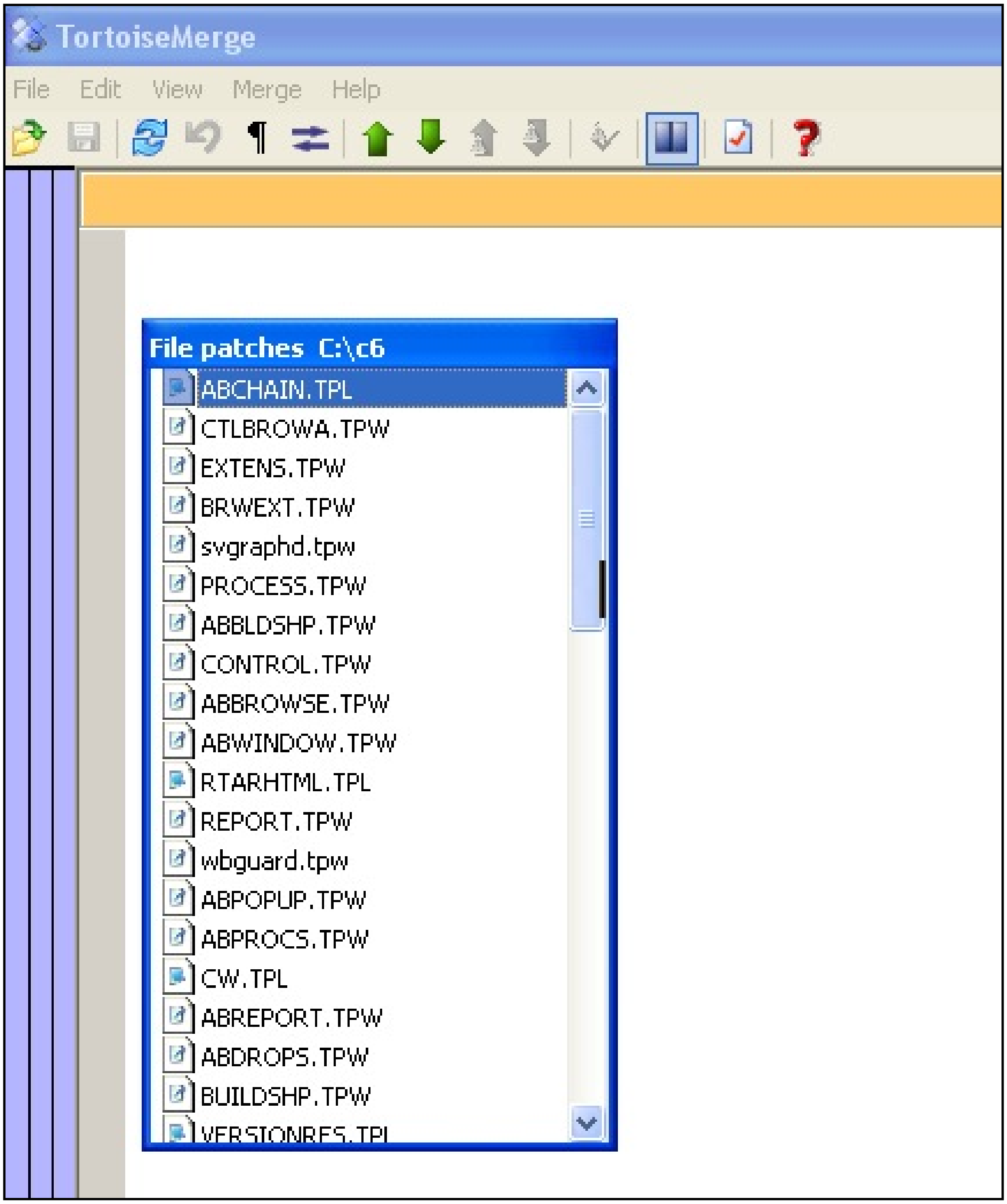


```
17   dharms   4:12:43 PM, 20 January, 2006   Build 9050 (6.3) - he
16   dharms   4:04:41 PM, 20 January, 2006   Build 9050 (6.3)
14   dharms   3:55:56 PM, 20 January, 2006   Build 9049
12   dharms                                  048
10   dharms                                  047
8    dharms                                  046
4    dharms

1 9049
```

Figure 2. Selecting the comparison method

From the context menu in Figure 2, choose Compare with working copy. TortoiseSVN will build a difference file, which may take a minute or two. When that process is done, you'll see the TortoiseMerge window, a portion of which you can see in Figure 3.

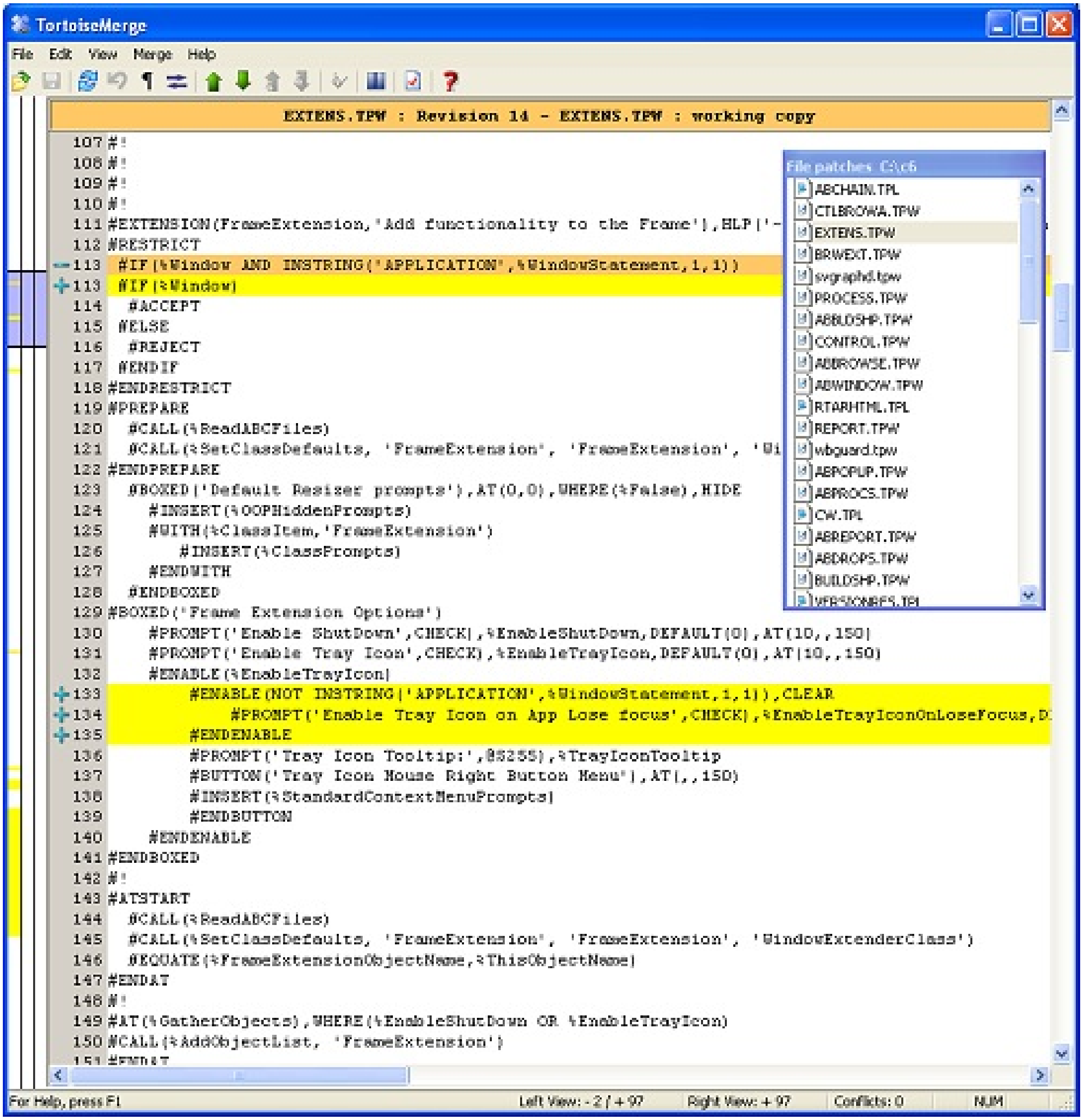


ABCHAIN.TPL
CTLBROWA.TPW
EXTENS.TPW
BRWEXT.TPW
svgraphd.tpw
PROCESS.TPW
ABBLDSHP.TPW
CONTROL.TPW
ABBROWSE.TPW
ABWINDOW.TPW
RTARHTML.TPL
REPORT.TPW
wbguard.tpw
ABPOPUP.TPW
ABPROCS.TPW
CW.TPL
ABREPORT.TPW
ABDROPS.TPW
BUILDSHP.TPW
VERSIONRES.TPL

Figure 3. The TortoiseMerge window

You can see a list box showing the changed files. This is actually the File Patches list (TortoiseMerge is a *merge* tool, after all), meaning that each of these represents a patch which you can apply to the file in question. I suggest you do *not* avail yourself of this functionality, as you really don't want to modify your Clarion working directory.

Instead, simply double-click on any file in the patch list to view the differences between the two versions. By default, TortoiseMerge will show you the changes in side-by-side mode, but you can toggle between side-by-side and inline mode by clicking on that little two-pane icon on the list, pressing Ctrl-D, or by choosing View|Two-way Diff.



EXTENS.TPW : Revision 14 - EXTENS.TPW : working copy

```
  #!
  #!
  #!
  #!
  #EXTENSION(FrameExtension,'Add functionality to the Frame'),HLP('-
  #RESTRICT
-  #IF(%Window AND INSTRING('APPLICATION',%WindowStatement,1,1))
+  #IF(%Window)
    #ACCEPT
   #ELSE
    #REJECT
   #ENDIF
  #ENDRESTRICT
  #PREPARE
    #CALL(%ReadABCFiles)
    #CALL(%SetClassDefaults, 'FrameExtension', 'FrameExtension', 'Wi
  #ENDPREPARE
    #BOXED('Default Resizer prompts'),AT(0,0),WHERE(%False),HIDE
      #INSERT(%OOPHiddenPrompts)
      #WITH(%ClassItem,'FrameExtension')
         #INSERT(%ClassPrompts)
      #ENDWITH
    #ENDBOXED
  #BOXED('Frame Extension Options')
      #PROMPT('Enable ShutDown',CHECK),%EnableShutDown,DEFAULT(0),AT(10,,150)
      #PROMPT('Enable Tray Icon',CHECK),%EnableTrayIcon,DEFAULT(0),AT(10,,150)
      #ENABLE(%EnableTrayIcon)
+         #ENABLE(NOT INSTRING('APPLICATION',%WindowStatement,1,1)),CLEAR
+             #PROMPT('Enable Tray Icon on App Lose focus',CHECK),%EnableTrayIconOnLoseFocus,D
+         #ENDENABLE
          #PROMPT('Tray Icon Tooltip:',@S255),%TrayIconTooltip
          #BUTTON('Tray Icon Mouse Right Button Menu'),AT(,,150)
          #INSERT(%StandardContextMenuPrompts)
          #ENDBUTTON
      #ENDENABLE
  #ENDBOXED
  #!
  #ATSTART
    #CALL(%ReadABCFiles)
    #CALL(%SetClassDefaults, 'FrameExtension', 'FrameExtension', 'WindowExtenderClass')
    #EQUATE(%FrameExtensionObjectName,%ThisObjectName)
  #ENDAT
  #!
  #AT(%GatherObjects),WHERE(%EnableShutDown OR %EnableTrayIcon)
  #CALL(%AddObjectList, 'FrameExtension')
  #ENDAT
```

For Help, press F1 | Left View: - 2 / + 97 | Right View: + 97 | Conflicts: 0 | NUM

Figure 4. Comparing versions (view full size image)

I prefer to see the changes in a single screen because it's easier to see all the code, but there are situations where side by side is clearer. The problem is that you often can't see all the code without scrolling horizontally. If you have dual monitors, however, you can stretch the window across both monitors, so one version is on the left monitor, and the other is on the right. This works best if both monitors are the same size and resolution.

> **TIP:** If you use TortoiseMerge, be aware that in version 1.3, and possibly other versions, when you switch to another file the line pointer is not reset, and just clicking on the Next Difference icon may not take you to the first difference. I found I had to manually select the first line (or any line before the first change) before the navigation buttons worked as expected.

Here are a few supplementary items which I gleaned from the template and source comparison:

- The `FrameExtension` template is no longer restricted to application frames
- New IP driver options added to `Report` and `Process` templates, `FileDrop` and `FileDropCombo` controls. There's a fair bit of code sprinkled throughout the templates and classes related to the IP driver and support for multiple request packets. See `MRPSize` and `PROP:IPRequestCount` in the C6.3 help file.
- DOS file lookup control initial directory field increased to `@S255`
- Some initialization changes to the WebGuard template, and the `WbGuardClass` now uses an `ErrorStatusClass`.
- Server autoinc coded added to CW.TPL
- The version resource template now lets you include Clarion version information
- `AppsKey` is now alerted in `#PrepareAlerts`. This is the context menu key on your keyboard, usually located between the Windows and Ctrl keys on the right side, and you can use it instead of a right-click. The code to support this key has been added to browse handling.
- Some changes to the `JOIN` creation template code, evidently to support secondary custom joins
- Some template changes in RELATION.TPW in the gather relations code, to prevent this happening when there are server side constraints
- EIP – a lot of changes. If you're deep into EIP (which in the past has been a bit of a hair-puller) you should look at this closely. There are new methods in `EditDropComboClass` include `GetQueueMatch`, `SetFrom`, `SetEntryCompletion`, and `TakeNewSelection`.
- Thumb locator changes
- Minor Enter/Tab changes

My vote for the most pervasive change goes to the IP driver MRP code. I lost count of how many different files had related code changes.

There are, of course, other changes in the runtime and drivers which aren't necessarily reflected in the source code and templates. But I still like to look at the code, if for no other reason than to appreciate the complexity and power of the Clarion code generation system.

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David produces the Planet Clarion podcast, which he co-hosts with Andrew Guidroz II.

Reader Comments

Add a comment

- » Dave, You mention you use SVN and Tortoise for version...
- » John, The basic syntax for accessing Subversion on the...

Clarion Magazine

Anyone Home on the Network?

by Steven Parker and Henry Plotkin

Published 2006-01-20

`IsNetworkAlive`, `IsDestinationReachable`, `GetLastError`, `FormatMessage`, `GetDriveType`, `WnetGetConnection` … the odyssey to know the status of the network began with a printer error. It was the most peculiar printer error either of us ever saw.

The Point of Sale (POS) system we work on typically has several cash register PCs. Each register has a printer attached to it. Shared files are on a file server. Sometimes the file server happens to be one of the registers, sometimes it is a real server on the corporate network, sometimes it is a PC in the backroom. Nothing very unusual about this.

Last year we started using Epson USB printers. Each is connected to a register PC just as before.

So far, totally normal. However, sometimes, if the network connection dies, the application goes into "not responding" mode and has to be terminated from TaskManager. When this sort of thing happens with parallel printers, on restoration of the network, the user powers the printer off and on and continues. With USB printers, the printer displays an error message. That message references an error code that is not in Epson’s documentation. And when this error code is thrown, the register has to be rebooted. Mind, this is after restoration of the connection.

Having to reboot a client to reset its printer when the network goes off to admire the sunset is definitely not normal. Not in any sense is this acceptable.

Our hardware hit man, Marty Portner, eventually deduced that the printer handle was no longer valid. And printer handles are needed for both parallel and USB printers.

Parallel printers: `WriteFile` is used to write directly to a port (the reader is referred to the definitive discussion of direct to port techniques by Paul Attryde in his Clarion Insider articles.) `WriteFile` requires a handle so, during app initialization, `CreateFile` call is used to return the handle.

USB printing: using Windows printing, like writing to a printer on a parallel or serial port, requires a handle. OpenPrinterA gets that handle when using Windows printers. So, as far as initializing the app is concerned, the type of printer doesn’t matter, a handle is required. The difference is in how we get that handle and that ought not be significant.

None of us could figure out how a handle to a local resource could be invalidated by a failed network connection. (Marty eventually demonstrated that it is probably a bug in the Epson driver.) But one thing seems obvious: If we could determine that the network had died, the handle could be closed before terminating the application. Even if the handle is invalid, closing the handle releases the lock Epson places on the resource and eliminates the need to reboot the client. We could work around the bug.

Thus the theory. And thus the need to know the network’s status.

We "only" need to be able to know that the network has stopped responding. Our adventures in determining the status of the network are detailed in four articles, starting with Henry’s Hello, Server? The only problem is that Steve discovered that neither `IsNetworkAlive` nor `IsDestinationReachable` did the job as expected. Both of these API could return "good" results even when the network cable had been physically removed from the client PC.

A network cannot get any more "down" than a disconnected cable. But Windows still thought all was well; both APIs returned "good" results. Paul Attryde recently observed: "This is why I don't like a lot of the network API's - they'll only tell you if it's reachable in theory, not whether you can get to it at that exact moment or not." He repeatedly noted how incredibly slow Windows is to time out (a claim we were later able to confirm).

This precisely echoes our experience. Relying on Windows’ APIs, it could take hours to get the expected "network not available" response.

Paul’s method of checking network status, as explained in a comp.lang.clarion posting in July 2004 and in articles on his web site, is:

1. Call `GetDriveType` to see if it's remote
2. Call `WNetGetConnection` to get the drive mapping
3. With the name, use winsock to resolve the mapping to an IP address (see his Winsock Initialisation and Termination and TCP/IP name and address resolution)
4. Ping the IP with ICMPEcho
5. If the ping works, use a file API (like `GetFileAttributes`) to see if you can touch the file

The logic behind this is that if the ping fails, you know the net is gone. Immediate action can be taken. But even if the ping succeeds, because Microsoft’s peer networking shares do degrade over time, double check by trying to touch an actual file. Again, Paul puts the "touch" as the last step because of the extreme slowness of Windows 95 and 98, which were the OSs current when he devised his strategy.

Neither of us (that is to say, Steve) wanted to get involved with the WinSock API. At least, not just yet. Why not just jump straight to step 5, touching a specific file on the server? (Steve says: that is what it means to be a lazy programmer.)

GetFileAttributes only requires a CString "that specifies the name of a file or directory." In fact, a function from CDD days, Access, is just that, a wrapper for `GetFileAttributes`.

This got us thinking about alternatives. Late in the Clarion 5 development cycle, Topspeed introduced the `Exists` function. It, too, looks like a wrapper for `GetFileAttributes`. It should do (except that in the Windows incarnations, it doesn’t support wildcards, as we found out later).

`Directory`, introduced in CDD, should also do. And, in Windows, `Directory` supports wildcards.

Going back even further, to CPD, Steve used the C `FindFirst` function to create an `Exists` function.

So, there are three possible ways to avoid delving into the WinSock API.

Exists

The prototype for Exists, from builtins.clw, is:

```
    EXISTS(STRING),LONG,NAME('Cla$FileExists')
```

`Exists` is built into Clarion. It takes a string containing the file to check for and returns true or false.

For example,

```
    If Exists('c:\clarion6\c6ee.exe')
      ! do something
    End
```

Directory

The Directory function is a little more complicated. But it works correctly with wildcards.

```
    DIRECTORY( queue, path, attributes )
```

The queue that is required must be of one of two structures:

```
    ff_:queue  QUEUE,PRE(ff_),TYPE
    name        STRING(13)
    date        LONG
    time        LONG
    size        LONG
    attrib      BYTE
               END
```

This queue supports short file names only. For long file names use:

```
    FILE:queue QUEUE,PRE(File),TYPE
    name        STRING(FILE:MAXFILENAME)
    shortname   STRING(13)
    date        LONG
    time        LONG
    size        LONG
    attrib      BYTE
               END
```

In practice, you only need a declaration like the following at an appropriate data embed:

```
    MyQueue     Queue(FILE:queue)
               End
```

The second parameter is just a string containing what to check for.

The final parameter, attributes, is selected/constructed from:

```
    ff_:NORMAL    EQUATE(0)    !Always active
    ff_:READONLY  EQUATE(1)    !Not for use as attributes parameter
    ff_:HIDDEN    EQUATE(2)
    ff_:SYSTEM    EQUATE(4)
    ff_:DIRECTORY EQUATE(10H)
    ff_:ARCHIVE   EQUATE(20H)  ! NOT Win95 compatible
```

In use, the code looks like this:

```
    Free(MyQueue)
    Pattern = Clip(DataPath) & 'optionFile.TPS'
    Directory(MyQueue, Pattern, ff_:NORMAL)
```

And, if the queue has records, the file(s) exist. Otherwise, they do not.

FindFirst

The C `FindFirst` we used in DOS becomes `FindFirstFile` in the Windows kernel.

> "The **FindFirstFile** function searches a directory for a file or subdirectory with a name that matches a specific name."

Sounds perfect.

This is the most complex of all as it is now a Windows API call (see FindFirstFile) and Redmond seems incapable of creating a simple call (Sleep is the exception proving the rule).

`FindFirstfile` takes a pointer to a `CString` containing the target of the test. It also takes a group structure. So, its prototype is:

```
    FindFirstFile( *CSTRING lpFileName, *WIN32_FIND_DATA ),
        LONG,RAW,PASCAL,NAME('FindFirstFileA')
```

In fact, the prototype and the `WIN32_FIND_DATA` structure are in cwutils.inc and .clw in the Clarion6 libsrc directory. Instead of making yourself crazy, grab what you need from there or copy it from the demo app.

You will find, however, that the `WIN32_FIND_DATA` structure refers to a `FileTime` group (also found in cwutils) three times. This causes compiler warnings. In the demonstration app at the end of the article, we (meaning Henry) have made a slight change; we do not inherit the `FileTime` group, using three real groups within the `WIN32_FIND_DATA` group. This stops the warnings.

`FindFirst`, in DOS, returns zero for true, anything else is false. `FindFirstFile` is a bit more complicated. If `FindFirstFile` fails to find the file or subdirectory, it returns `INVALID_HANDLE_VALUE` (-1). If it succeeds, it gives the handle to the file(s).

So, the final test looks like this:

```
Pattern = System{Prop:DataPath} & '\optionFile.TPS'
If FindFirstFile(Pattern, MyGroup) > 0
  ! found it
Else
  ! file doesn't exist or network down
End
```

The Proof of the Pudding ...

There are two things we need to know about each of the three possible techniques:

1. Do they return results as expected?
2. Do they return results in a reasonable timeframe?

The downloadable app shows the code implemented. If you do not have Clarion 6.2, a locally compiled EXE is provided for you to test.

The app starts by asking for a file to check. We tested with local paths, UNC paths and paths using mapped drives. We tested with files that existed and files that didn’t. Testing was done on a wireless, 11Mb network (i.e., a slow net). The first series was run with the network up; the second was run with the wireless network card pulled on one of the machines. Tests were run with and without wild cards.

In all cases, except wildcards, `Exists`, `Directory` and `FindFirstFile` gave the correct results. With wildcards, `Exists` always returned "not found." You can easily verify this for yourself.

`Exists` is easiest but looking less appetizing. It is entirely inappropriate for testing with wild cards.

Timing, measured in Clarion time (1/100th second) varied wildly.

	Run 1	**Run 2**	**Run 3**	**Run 4**
`Exists`	3	4	4	4
`Directory`	3	5	3	4
`FindFirstFile`	2	1	2	1

Another test series resulted in these values:

	Run 1	**Run 2**	**Run3**	**Run 4**
`Exists`	142	4	4	4
`Directory`	1	6	6	6
`FindFirstFile`	1	3	3	4

The results when the card was pulled where somewhat more consistent:

	Run 1	**Run 2**	**Run 3**	**Run 4**
`Exists`	2523	228	3	228
`Directory`	225	0	3	0
`FindFirstFile`	0	0	1	0

The second test series was similar:

	Run 1	**Run 2**	**Run 3**	**Run 4**
`Exists`	2504	229	227	227
`Directory`	2103	0	0	5
`FindFirstFile`	0	0	0	0

Note that there is a tendency for results to return more quickly the more often the test program is run. At least this is our impression from playing around with additional test sequences. Also, these timing include the time to clear the passed queue or group.

These results completely eliminate `Exists` from the competition.

Summary

For casual use, `Exists` is excellent. It does not need to be prototyped and the call is totally straightforward. But for serious network testing, for example in the `Event:Timer` of a browse, 25 seconds when the network goes off for R & R, `Exists` is unacceptable.

`Directory` and `FindFirstFile` both return correct results in an acceptable timeframe. `Directory` is more familiar to most Clarion programmers and may be used for other purposes in an app.

For total performance, however, `FindFirstFile` is consistently quicker than the other methods, especially when there is a network failure. And it is reliable. It is going into our app.

Most remarkable in the *sturm und drang* of the past two weeks is the final outcome. After investigating and testing all these APIs, after what seems like countless discussion, after the generous hand-holding by other developers, we end up settling on code written 15 years ago.

In DOS.

Now, ain't that a pip?

Download the source

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

"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

Clarion Magazine

Next Month Anniversary Date Function

by John L Griffiths

Published 2006-01-19

Reading the recent article by Carl Barnes has prompted me to put pen to paper (well, I really mean fingers to keyboard).

Often I need to advance a date by one month, but the date in the next month must be based on the anniversary of an original date. In my case, this is the date an account was opened. The standard `DATE()` function works fine if the account opened date lies between the 1st and 28th day of the first month. But what if an account opened on January 30 2005? Adding one month each time to the previous in the series would give a dates of sequence of Jan 30, March 2 April 2 May 2. Where did the February date go?

So I built a very simple small function I call `nmDate` to give me the date for the next month. I require a date for each month. I pass two values and the function returns the next month’s date as a `LONG`.

Here is the declaration:

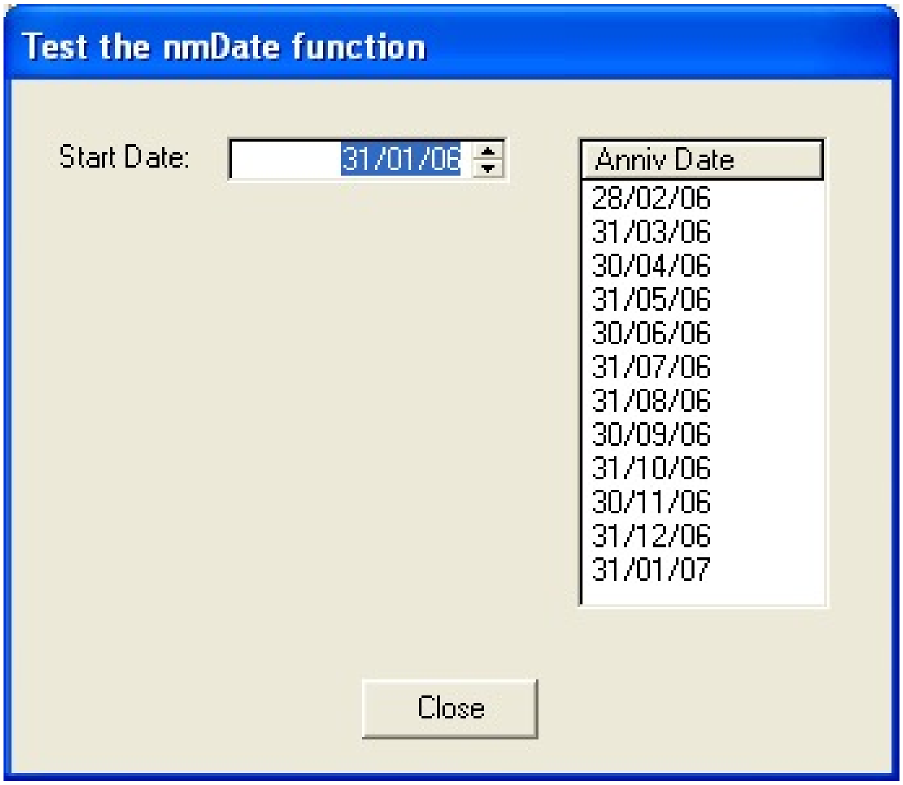
```
nmDate(LONG piFromDate,LONG piAnnivOf),LONG
```

And here's the code:

```
nmDate     FUNCTION (LONG piFromDate,LONG piAnnivOf) !,LONG
 ! piFromDate is the date in the month which we want to
 ! advance and get the Next Months date. The piAnnivOf is
 ! the day of month value we wish to use as the anniversary
 !
NRD     LONG,AUTO
```

```
CODE
CASE piAnnivOf
OF 1 to 28
  NRD = date(MONTH(piFromDate)+1, piAnnivOf ,year(piFromDate))
OF 29 to 31
  NRD= date(MONTH(piFromDate)+1, piAnnivOf ,year(piFromDate))
  IF DAY(NRD) < 5
    LOOP 4 TIMES
      NRD -= 1
      IF DAY(NRD) > 27
        BREAK
      END
    END
  END
ELSE
  NRD = 0
END ! CASE
RETURN NRD
```

Now, using `nmDate` and a starting date of Jan 31, I get the sequence of Feb 28, March 31, April 30, etc. as shown in Figure 1.



Start Date: 31/01/06

Anniv Date
28/02/06
31/03/06
30/04/06
31/05/06
30/06/06
31/07/06
31/08/06
30/09/06
31/10/06
30/11/06
31/12/06
31/01/07

Figure 1. Generating anniversary dates

Download the source

John Griffiths, an Australian, has been developing with Clarion since the DOS days. John has two summers each year, spending six months in Australia for the Southern summer and six months in Texas for the Northern summer. He works as a contractor for a Texas company and has developed several business/financial programs which he sells worldwide. John has a B.Bus degree with a major in Informations Systems.

Reader Comments

Add a comment

- » This is great... I have been wondering how to correct all...
- » You could always get the first date of the next month and...
- » But we are not always looking for the last day of...

Clarion Magazine

Edit-In-Place: Getting User Confirmation

by Henry Plotkin

Published 2006-01-13

No good deed, it is said, goes unpunished. And I now come to realize that edit-in-place is ready, willing and able to punish the good-deed-doing Clarion developer. Perfectly willing.

My first serious exposure to edit-in-place is described in When Edit-In-Place Bites Back. The problem therein discussed involves physical inventory worksheets created by batch selecting data directly from the inventory file.

A worksheet can be transmitted to one or more scanners so clerks can wander up and down aisles, pointing their scanners and counting things. The data in the scanners can then be read from the scanner to update the worksheet. Multiple scanners present no issue. Problems begin only when users manually update the on hand quantity.

One particular user would start to edit the on hand for an item and, without completing the field, would click the change button again. The new quantity displayed in the browse. But embedded code to update inventory did not execute. It was later demonstrated to me that clicking anywhere in the list box would have the same consequence. A resolution was presented in When Edit-In-Place Bites Back Some More.

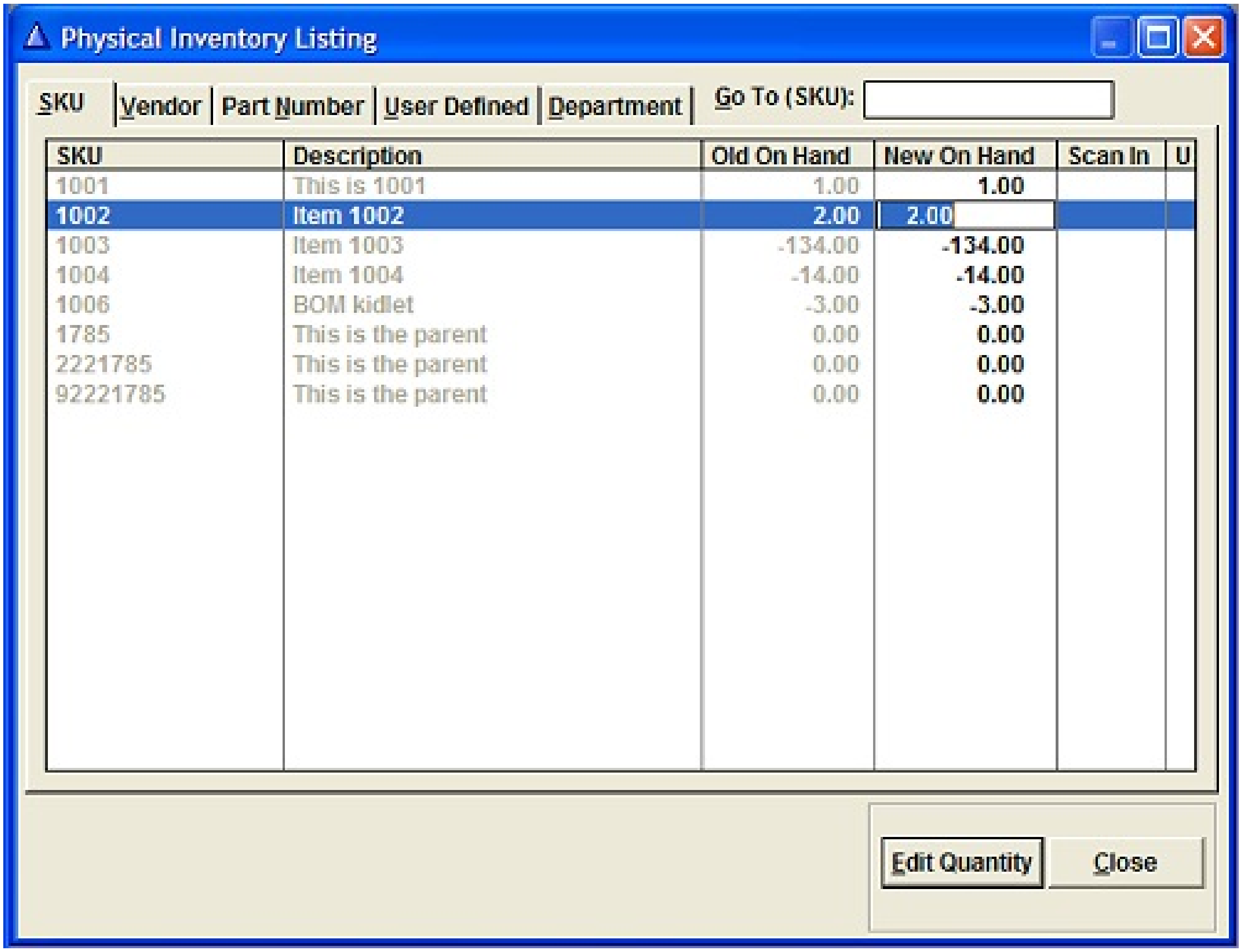
Ultimately, updating inventory for manual changes was integrated with updates received from the scanners. This updating is done by a batch process on the parent browse. All my work is obviated because manual changes are no longer reflected in real time.

Users, being sometimes too clever for my own good, however, found a new way to ball the works up. Several users apparently keep their inventory both on the sales floor and in a stock room. They scan their inventory on the sales floor but count it by hand in the stock room.

To update the physical inventory worksheet, some of these users do their manual adjustments first and their scanned adjustments second. In this case, the software correctly adds the scanned quantity to any pre-existing quantity.

However, some users import the quantities from the scanner first and do their manual adjustments afterwards. In this case, they become confused.

Suppose the new on hand from the scanner puts two (2) in the browse, as in Figure 1.



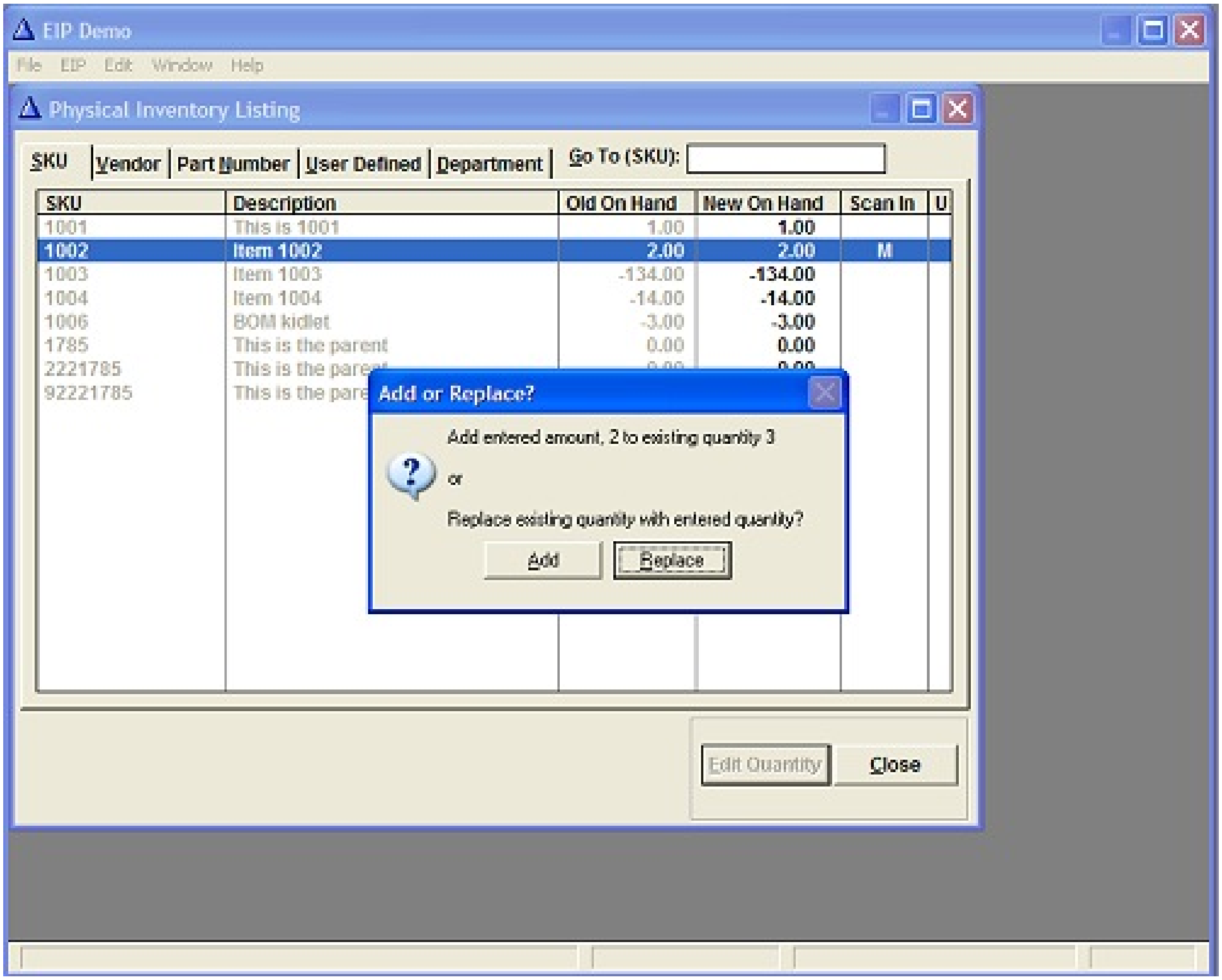
SKU	Description	Old On Hand	New On Hand	Scan In	U
1001	This is 1001	1.00	1.00		
1002	Item 1002	2.00	2.00		
1003	Item 1003	-134.00	-134.00		
1004	Item 1004	-14.00	-14.00		
1006	BOM kidlet	-3.00	-3.00		
1785	This is the parent	0.00	0.00		
2221785	This is the parent	0.00	0.00		
92221785	This is the parent	0.00	0.00		

Figure 1. Manual adjustment with existing quantity (see full size image)

Suppose the manual count for item is seven (7). What is the user to enter? Nine (9) seems the obvious answer. But, as stated, users are clever. Some of them do indeed want to add the existing quantity to the quantity manually counted. Some, for reasons beyond anyone’s comprehension, want to replace the existing amount with the manual count (perhaps the manual count already includes the pre-existing amount; who knows?). Of course, in the first case, the user must also do the addition themselves. (This, of course, is an opportunity for error and, more importantly, a tech support call.)

The software is good. But it cannot divine the user's intention.

But, I *can* ask the user what he wants to do. If there is no pre-existing quantity, whatever the user enters is the correct value. If there is an existing value, any value, ask the user whether he wants to add or to replace. Something like:



SKU	Description	Old On Hand	New On Hand	Scan In	U
1001	This is 1001	1.00	1.00		
1002	Item 1002	2.00	2.00	M	
1003	Item 1003	-134.00	-134.00		
1004	Item 1004	-14.00	-14.00		
1006	BOM kidlet	-3.00	-3.00		
1785	This is the parent	0.00	0.00		
2221785	This is the parent	0.00	0.00		
92221785	This is the pare				

Add entered amount, 2 to existing quantity 3

or

Replace existing quantity with entered quantity?

Figure 2. Getting user confirmation of an EIP change (see full size image)

Sounds simple.

First, I need to know what the new on hand value is when the browse record (row) is selected, i.e. before any edits. Then I need to know when the user has completed data entry. Finally, I can check whether the initial value of new on hand was zero or not; if non-zero, then I can check whether the current value of the control is the same as the previous value.

Getting and saving the previous value of a column in a browse *is* easy. In the browse's `TakeNewSelection` event/embed:

```
SaveQty = PID:NewOnhand
```

I want to ask the user any time `SaveQty` is non-zero or any time `SaveQty <> PID:NewOnhand`.

I also know that if `SaveQty = 0`, I do not need to ask. Therefore, my conditions will look something like:

```
If PID:NewOnHand <> SaveQty or (SaveQty <> 0) ! changed?
  If SaveQty <> 0   !ask only if initial value exists
    CASE MESSAGE('Add entered amount, ' & PID:NewOnHand & |
         ' to existing quantity ' & SaveQty & '||or' &|
         '||Replace existing quantity with entered ' & |
         'quantity?','Add or Replace?', ICON:Question, |
         '&Add|&Replace', 2, 0)
    OF 1  ! Name: &Add  (Default)
      PID:NewOnHand = PID:NewOnHand + SaveQty
      BRW1.Q.PID:NewOnHand = PID:NewOnHand
    OF 2  ! Name: &Replace
    END !CASE
  End
  BRW1.Q.PID:Scanned = 'M'      !update "action" display
  PID:Scanned = BRW1.Q.PID:Scanned
  PID:NEWDATE = Today()
  Put(PIDETAIL)
End
```

Unfortunately, EIP controls do not post events typically associated with either browse or entry controls. These include `Event:Selected`, `Event:Accepted` and the like. Without these events, there are none of the embeds normally associated with them.

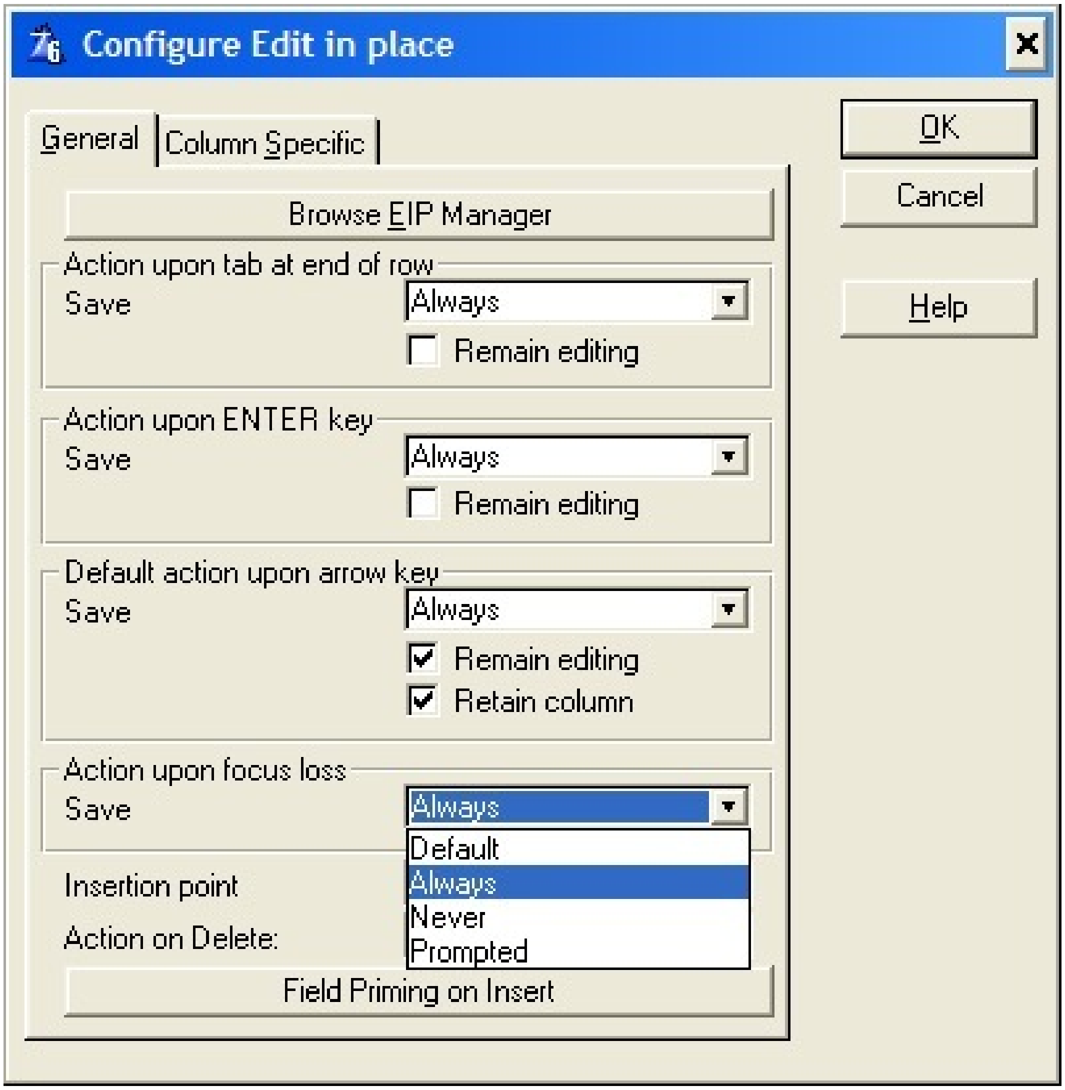
Therefore, little pertaining to edit-in-place is simple. The evidence for this is to be found in the number of articles on the subject at Clarion Magazine as well as the number of FAQs in the various knowledge bases. In particular, without an Event:Accepted (and attendant embed), I have no ready place to make my tests. I have no place to embed the code shown above.

I work for "Mr. Ignore the Templates" (a.k.a Steven Parker). So, throwing up my hands in despair is not an option.

In fact, edit-in-place is not a template. It is a class. Therefore, there is no template to ignore. Throwing up my hands in despair? Still not an option.

The embeds I need, especially the `Accepted` embed, are not present because entry fields and edit-in-place controls are completed differently. A normal entry field can be completed in a very limited number of ways, typically only by the enter or tab keys. Normal entry fields do not normally complete if no changes are actually made to the field contents. Edit-in-place controls can be completed in a number of ways: arrow up, arrow down, left arrow, right arrow, tab, enter, even change of focus. Edit-in-place controls complete based on keys, not on changes to contents.

Making edit-in-place even more complex, the developer can select how the EIP Field Managers for a procedure (remember that edit-in-place controls, therefore objects, are created on a control by control basis and, then, only when edit-in-place is explicitly activated for that field but the behaviors are procedure-wide). Figure 2 shows this.



Action upon tab at end of row
Save: Always
☐ Remain editing

Action upon ENTER key
Save: Always
☐ Remain editing

Default action upon arrow key
Save: Always
☑ Remain editing
☑ Retain column

Action upon focus loss
Save: Always

Insertion point

Action on Delete:

Figure 3. Edit-in-place, key by key options for a single field

In other words, edit-in-place controls can be set up to save contents on one key stroke, do nothing on another and ask on a third. That there are so many options indicates that there must be a mechanism for determining how the object is to act (or not act) on any given key stroke.

Indeed, because edit-in-place is implemented as a class, it logically follows that there must be ways to override its methods. Luckily, edit-in-place is fairly well documented, at a conceptual level. This documentation is found in the C5 Learning Clarion tutorial, starting at page 146 and the meat of the information begins on page 153 (editor's note: look on your Clarion CD for the Learning Clarion PDF; you can also order a printed version from SoftVelocity).

Here I learn that each edit-in-place control has a `TakeEvent` method (typically labeled `EditInPlace::<fieldLabel>.TakeEvent`). This method is like a standard entry control's All Events embed but it also returns a value from its parent call.

A bit of digging in `abeip.inc` reveals the values that can be returned:

```
EditAction ITEMIZE(0),PRE
None     EQUATE
Forward  EQUATE     ! Next field
Backward EQUATE
Complete EQUATE     ! OK
Cancel   EQUATE
Next     EQUATE     ! Next record
Previous EQUATE
Ignore   EQUATE
  END
```

So, `ReturnValue = EditAction:None` means the user did nothing, `EditAction:Cancel`, the user cancelled. `EditAction:Forward` is arrow right and `EditAction:Next`, arrow down. Etc. So, I need only wrap my code in

```
If ReturnValue and ReturnValue <> EditAction:Cancel
```

exactly as show in the tutorial (page 155) in the `TakeEvent` method for my code to execute precisely when and how I want it to. The final implementation becomes:

```
Update(Self.FEQ)  !update the EIP control
If ReturnValue and ReturnValue <> EditAction:Cancel
  If BRW1.Q.PID:NewOnHand <> SaveQty or (SaveQty <> 0)
    If SaveQty <> 0
      CASE MESSAGE('Add entered amount, ' & |
        BRW1.Q.PID:NewOnHand & ' to existing quantity '&|
        '' & SaveQty & '||or'&|
        '||Replace existing quantity with entered quantity?', |
        'Add or Replace?', ICON:Question,  |
        '&Add|&Replace', 2, 0)
      OF 1  ! Name: &Add  (Default)
        PID:NewOnHand = BRW1.Q.PID:NewOnHand + SaveQty
        BRW1.Q.PID:NewOnHand = PID:NewOnHand
      OF 2  ! Name: &Replace
      END !CASE
    End
    BRW1.Q.PID:Scanned = 'M'
    PID:Scanned = BRW1.Q.PID:Scanned
    PID:NEWDATE = Today()
    Put(PIDETAIL)
```

```
        End
        ?Change:2{Prop:Disable} = False
      End
```

Note that I must use the EIP fields in this variant, fields like BRW1.Q.PID:NewonHand. This is the other very important thing I picked up from the tutorial: the edit-in-place object creates new controls, one for each column where edit-in-place is enabled. It is this control, a control I did not create, that is managed by the EIP object and which the edit-in-place classes synchronize with the underlying view and browse fields. The tutorial clearly shows how and when I need to do this myself. (See the UpdatePIDetail2 procedure in the sample application.)

I am not sure I have ever seen these additional fields (e.g., BRW1.Q.<fieldLabel>) documented or described. It is by inference that I discover them (that and a quick look at the source files). Of course, were I truly ambitious, I could work my way through abeip.clw to get a thorough look at how edit-in-place is actually implmented (e.g., the EditClass INIT method). Nah, too much like work...

Alternately

EditInPlace::PID:NEWONHAND.TakeEvent PROCEDURE(UNSIGNED Event) is not exactly the first place I look when I want to act on a list box action. Usually, I use the browse's ResetFromAsk method. Indeed, in the previous articles, it is this embed that is used to update inventory.

The virtue of this embed is that not only is it one I am used to using but I can also use the file fields directly. Instead of queue fields like BRW1.Q.PID:NewOnHand, I can use real file fields like PID:NewOnHand, as usual.

The only problem is that the value of the edit action is local to TakeEvent. It is out of scope in ResetFromAsk. Of course, I can save the ReturnValue in TakeEvent and use it in ResetFromAsk. The UpdatePIDetail procedure in the sample application shows this way of implementing a user confirmation.

In conclusion

In the current case, there is no template to ignore. And that is a good thing. Because edit-in-place is implemented as an object for each control where edit-in-place is desired, there are just enough methods for me to override that I can accomplish useful work.

For example: note the `INIT` method for the entry control, where I ensure that the control is in insert mode. This is very similar to creating controls on the fly and having to set all the control's properties before displaying it.

In fact, this is the big lesson that I took away from this exercise: edit-in-place controls may have a template wrapper but one is best advised to consider them as if they had been created at runtime using the `Create` (return new control created) statement. A quick examination of the source show that this is precisely what is happening.

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

Clarion Magazine

Completely Dynamic Listbox Formatting?

by Steven Parker

Published 2006-01-11

The Point-of-Sale (POS) application I manage allows users to select subsets of inventory for a variety of purposes. These include order recommendations, physical inventory, creating store-wide sales and several other things. What started off as a reporting feature eventually became a browse requirement. And that meant creating not just dynamic filters, but dynamic list box formats.

I gather the user's selections using the window in Figure 1.

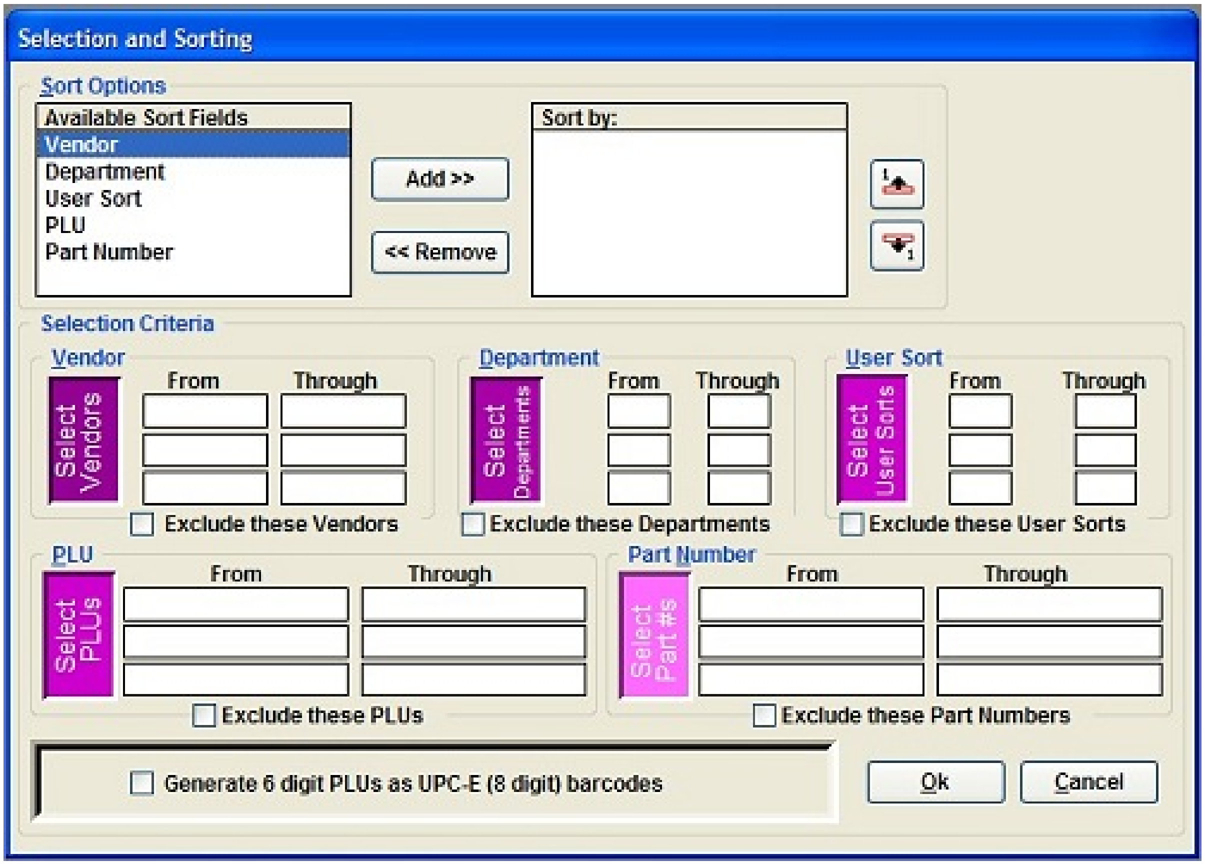


Figure 1. User selection window (see full size image)

In the middle of the window, you will notice a group titled "Selection Criteria" (Figure 2, below). Users may select inventory records by vendor, department, user sort, PLU (‘Price Lookup Unit," usually the UPC Code), part number *or any combination* of these five. They may enter or choose (each entry field calls a data validation lookup) a single value by making an entry in the left column. They may ask for a range of values by making entries in paired left and right columns. They may elect to include everything except the entered values by clicking the "Exclude" button at the bottom of each sub-group. They may combine excluded items/ranges with included ones.

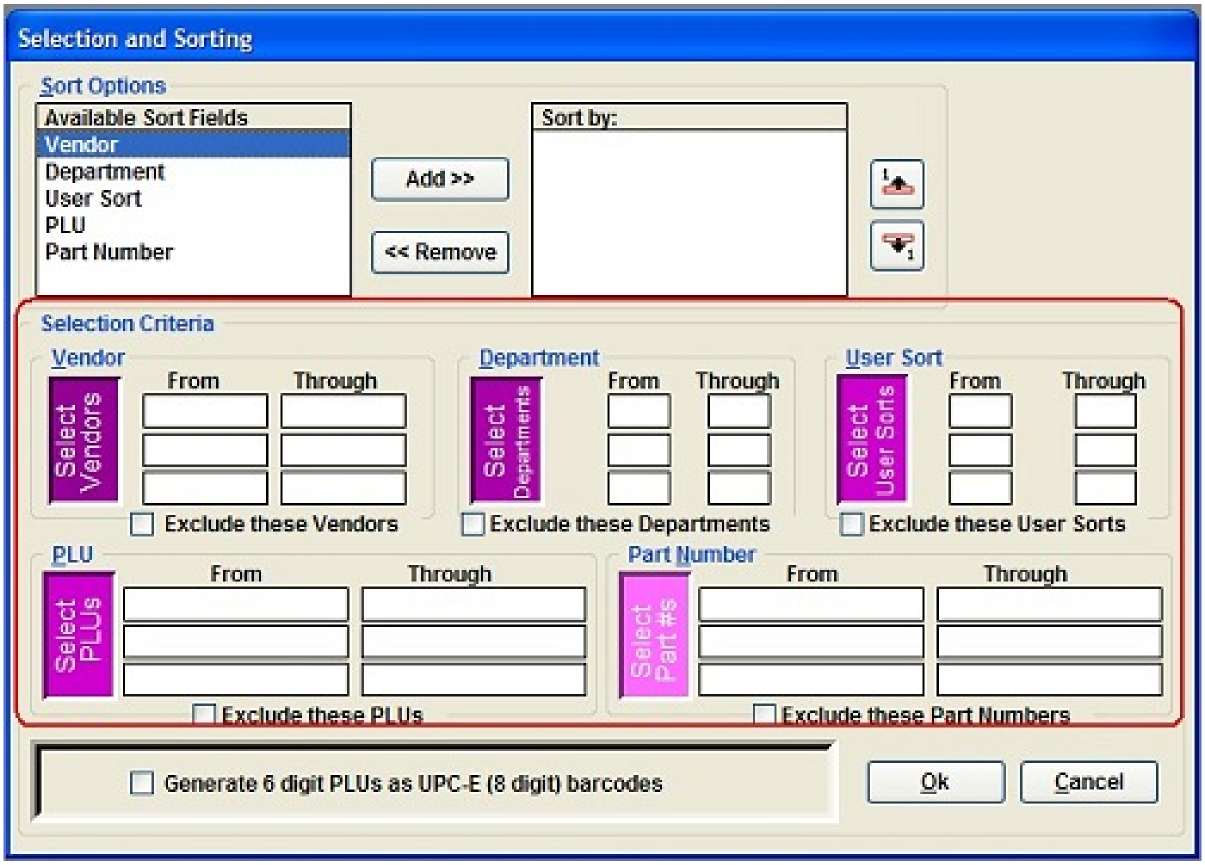


Figure 2. Selection Criteria entry area (see full size image)

When the user is done and clicks "Ok," embedded code creates a series of up to five strings. One string is created for each sub-group in which the user has made an entry. Each string contains a Clarion statement for use in a .SetFilter call.

Very flexible. Very well received by users (it replaces a DOS procedure that walks the user through seven screens, screen by screen, the user having to complete every screen in the sequence whether or not s/he wanted to make an entry).

There is also a "Sort Options" group (Figure 3, below) at the top of this window. Here, the user optionally creates a datum indicating how to sort items (this has less to do with the order in which inventory is ordered during selection than with how it is handled later). When this procedure opens, I load the "Available Sort Fields" queue manually. The "Add" and "Remove" buttons allow the user to move entries to/from the "Sort by" queue. The up and down arrow buttons let the user rearrange entries in the "Sort by" queue.

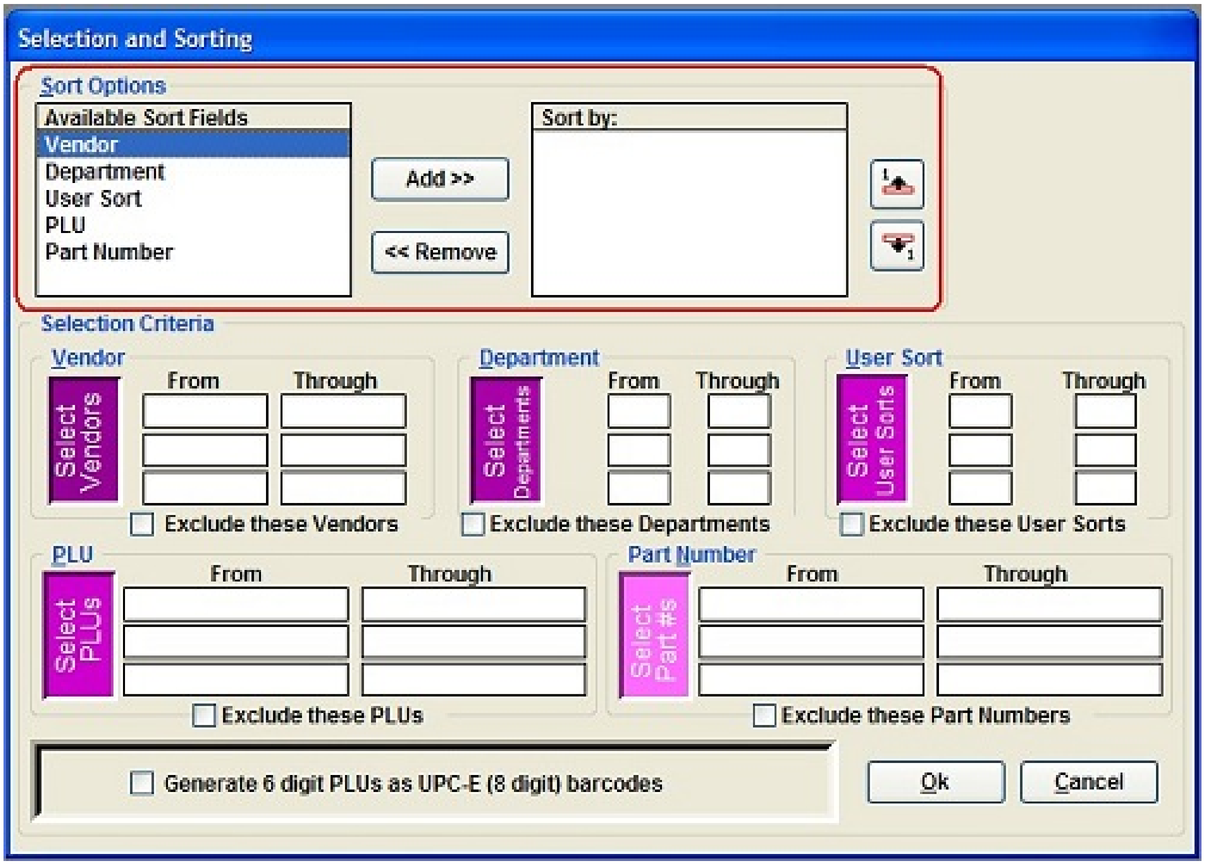


Figure 3. Sorting selection area (see full size image)

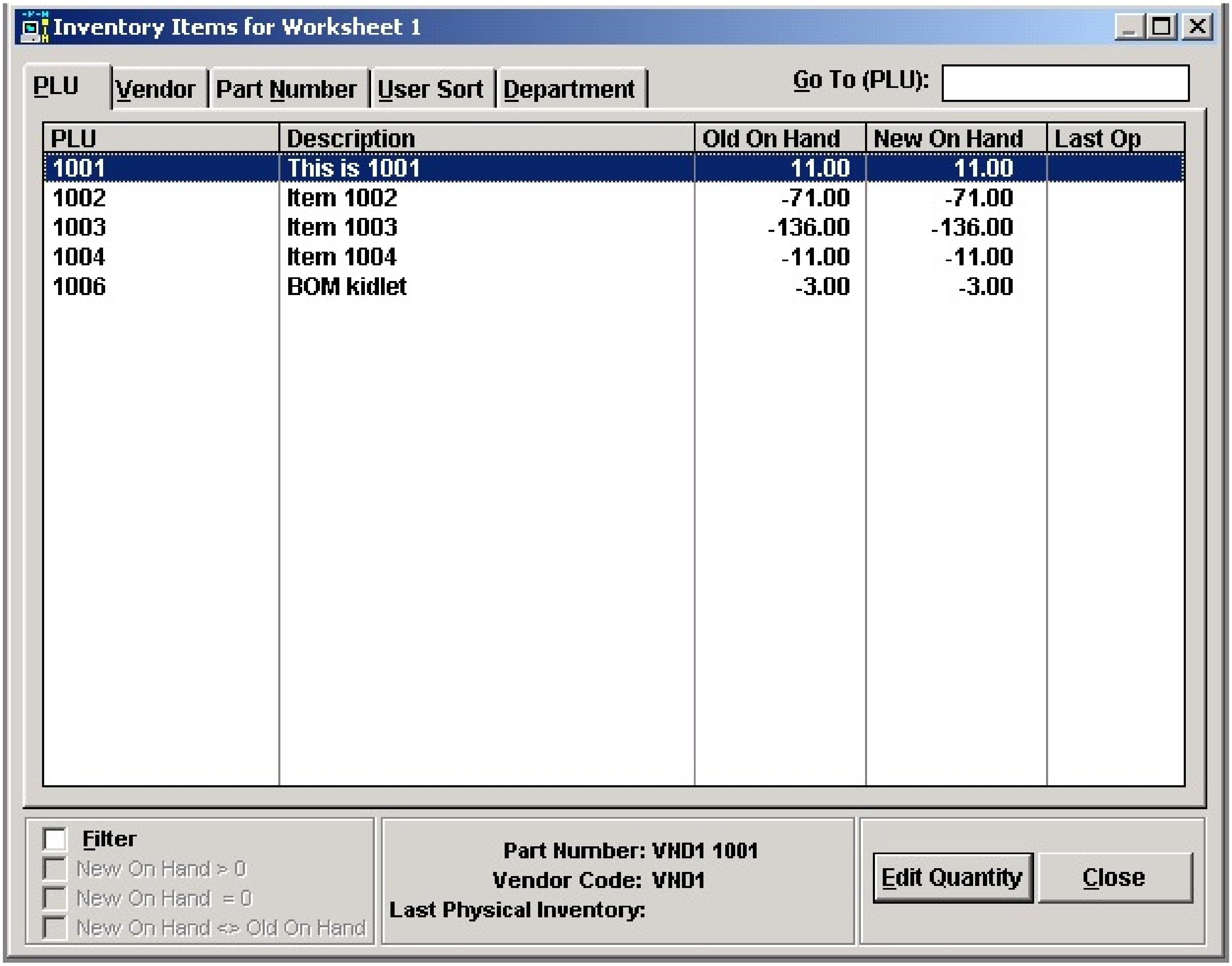
When the user is done and clicks "Ok," embedded code creates a string containing sufficient information for me to reconstruct the desired sort order whenever needed (a five character string with one letter for each selection). This datum is used in reports to construct a variable used in `SetOrder` and `AppendOrder` calls.

I believe I have described this window, focusing on how it primes the `SetFilter`, `SetOrder` and `AppendOrder` calls, in a number of previous articles. In fact, those articles were about filtering and ordering reports.

So how does this general purpose, if less than aesthetically compelling, window come back to haunt me? Well, it seems that the files created by these batch processes are not used only in reports. Some are used in browses. "Who’d a’ thunk it?"

When sort orders are used in browses, my users expect the classic Clarion multi-tabbed browse. They expect to be able to view their recommended orders or physical inventory by PLU, part number, vendor, department, etc.

I provide this. See Figure 4.



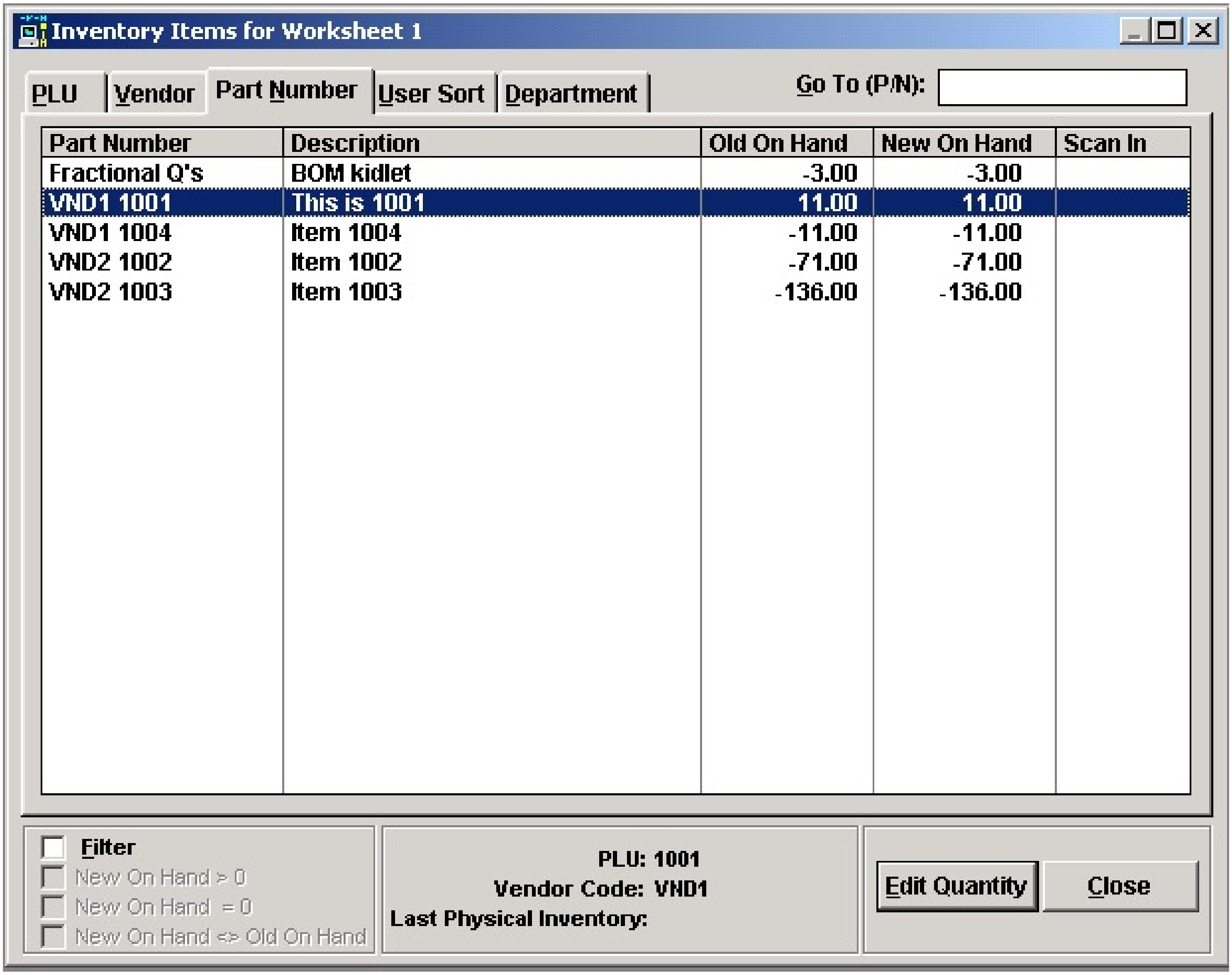
PLU	Description	Old On Hand	New On Hand	Last Op
1001	This is 1001	11.00	11.00	
1002	Item 1002	-71.00	-71.00	
1003	Item 1003	-136.00	-136.00	
1004	Item 1004	-11.00	-11.00	
1006	BOM kidlet	-3.00	-3.00	

☐ Filter
☐ New On Hand > 0
☐ New On Hand = 0
☐ New On Hand <> Old On Hand

Part Number: VND1 1001
Vendor Code: VND1
Last Physical Inventory:

Figure 4. Classic multi-tabbed browse

I even display the "sort field" first (Figure 5).



Part Number	Description	Old On Hand	New On Hand	Scan In
Fractional Q's	BOM kidlet	-3.00	-3.00	
VND1 1001	This is 1001	11.00	11.00	
VND1 1004	Item 1004	-11.00	-11.00	
VND2 1002	Item 1002	-71.00	-71.00	
VND2 1003	Item 1003	-136.00	-136.00	

□ Filter
□ New On Hand > 0
□ New On Hand = 0
□ New On Hand <> Old On Hand

PLU: 1001
Vendor Code: VND1
Last Physical Inventory:

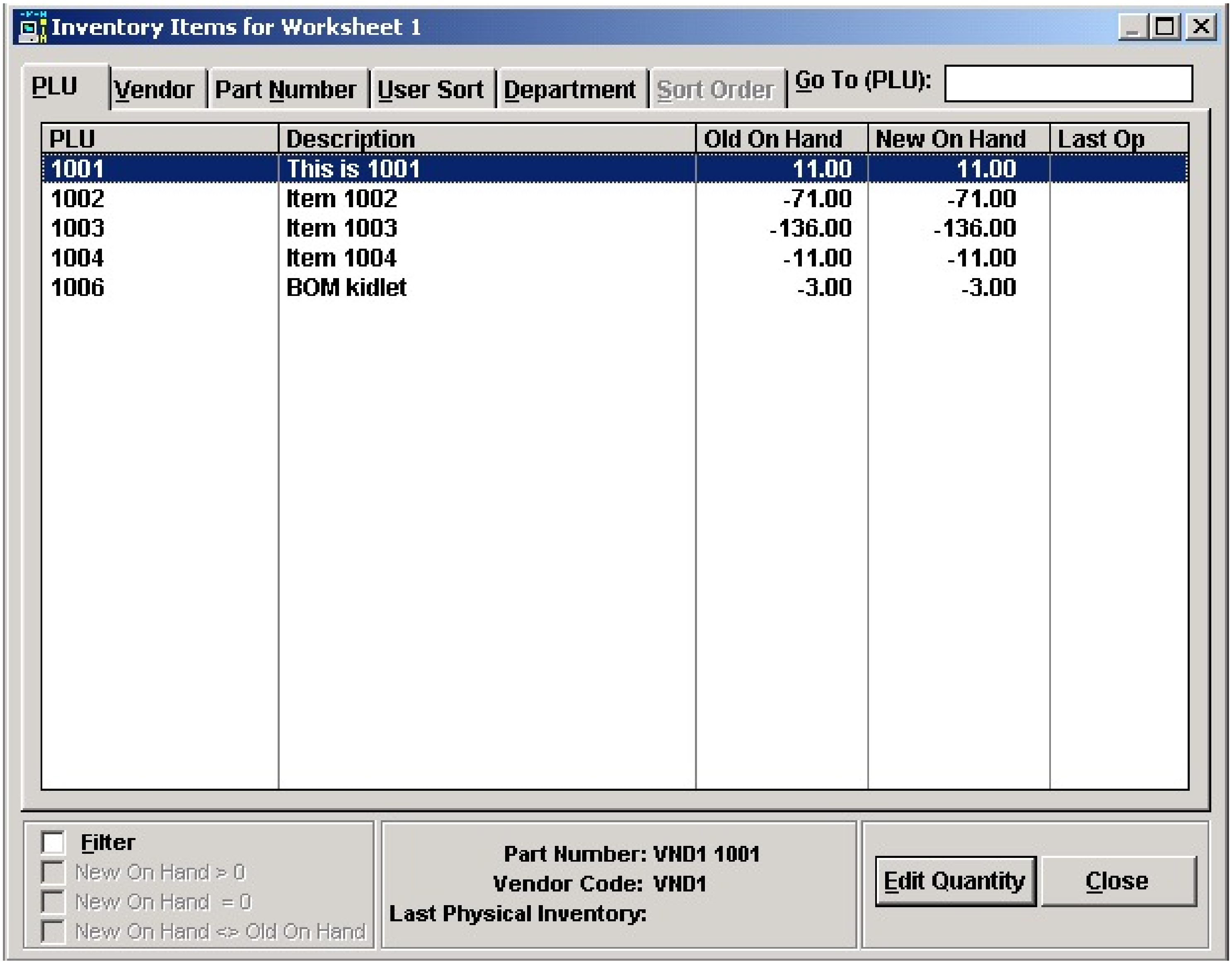
Figure 5. Sort Field first

Whenever the user changes tabs, the field named on the tab appears leftmost in the browse box with appropriate locator. In fact, if you compare Figures 4 and 5, you will note that there can even appear to be different columns in the different views. Actually, the "missing" columns’ width is set to zero; all of these columns are still there (but that’s another story).

Now, of all the unreasonable things, if the user made a selection in "Sort by," they want a tab showing the items in that order! Showing the file in the user defined order is the easy part but it is not all the users want. They want all the columns selected in "Sort by" to display!

If the user selects Part Number and Department, s/he wants to see those columns as the first two columns. But the Part Number tab, by default, shows Part Number and Description in the example procedure (refer to Figure 5, above).

If no sort order was selected, the new tab is inappropriate:



Go To (PLU):

PLU	Description	Old On Hand	New On Hand	Last Op
1001	This is 1001	11.00	11.00	
1002	Item 1002	-71.00	-71.00	
1003	Item 1003	-136.00	-136.00	
1004	Item 1004	-11.00	-11.00	
1006	BOM kidlet	-3.00	-3.00	

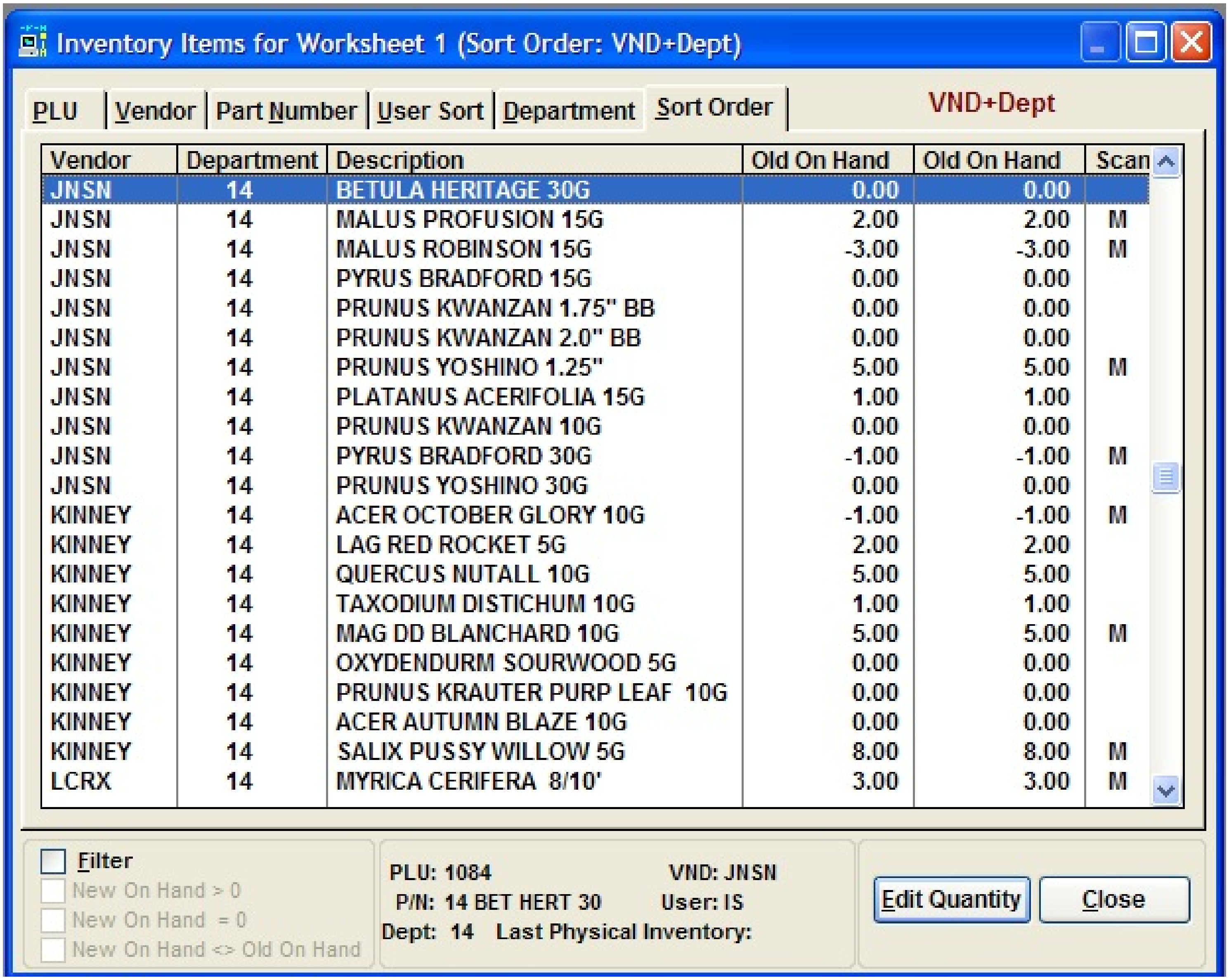
□ Filter
□ New On Hand > 0
□ New On Hand = 0
□ New On Hand <> Old On Hand

Part Number: VND1 1001
Vendor Code: VND1
Last Physical Inventory:

Figure 6. No "Sort by"

Similarly, if only one sort node is chosen, it must be the same as an existing tab. In this case, too, the new tab is inappropriate.

But, if a multivariate sort was selected, let the user see it in all its glory:



Vendor	Department	Description	Old On Hand	Old On Hand	Scan
JNSN	14	BETULA HERITAGE 30G	0.00	0.00	
JNSN	14	MALUS PROFUSION 15G	2.00	2.00	M
JNSN	14	MALUS ROBINSON 15G	-3.00	-3.00	M
JNSN	14	PYRUS BRADFORD 15G	0.00	0.00	
JNSN	14	PRUNUS KWANZAN 1.75" BB	0.00	0.00	
JNSN	14	PRUNUS KWANZAN 2.0" BB	0.00	0.00	
JNSN	14	PRUNUS YOSHINO 1.25"	5.00	5.00	M
JNSN	14	PLATANUS ACERIFOLIA 15G	1.00	1.00	
JNSN	14	PRUNUS KWANZAN 10G	0.00	0.00	
JNSN	14	PYRUS BRADFORD 30G	-1.00	-1.00	M
JNSN	14	PRUNUS YOSHINO 30G	0.00	0.00	
KINNEY	14	ACER OCTOBER GLORY 10G	-1.00	-1.00	M
KINNEY	14	LAG RED ROCKET 5G	2.00	2.00	
KINNEY	14	QUERCUS NUTALL 10G	5.00	5.00	
KINNEY	14	TAXODIUM DISTICHUM 10G	1.00	1.00	
KINNEY	14	MAG DD BLANCHARD 10G	5.00	5.00	M
KINNEY	14	OXYDENDURM SOURWOOD 5G	0.00	0.00	
KINNEY	14	PRUNUS KRAUTER PURP LEAF 10G	0.00	0.00	
KINNEY	14	ACER AUTUMN BLAZE 10G	0.00	0.00	
KINNEY	14	SALIX PUSSY WILLOW 5G	8.00	8.00	M
LCRX	14	MYRICA CERIFERA 8/10'	3.00	3.00	M

☐ Filter
☐ New On Hand > 0
☐ New On Hand = 0
☐ New On Hand <> Old On Hand

PLU: 1084 VND: JNSN
P/N: 14 BET HERT 30 User: IS
Dept: 14 Last Physical Inventory:

Figure 7. What the user specified

Providing "sort field first" is labor intensive. But it is not rocket science. SoftVelocity provides complete and detailed instructions on how to do this via the FORMAT string. In the on-line help, click FAQs | BrowseBoxes and LISTs | How to Display the Sort Field First on a Multi-Key Browse for the details.

Do *not* continue reading this article without reading the SoftVelocity FAQ.

The key to "sort field first" is:

- Parsing the `Format` string defining the list
- Adding the queue numbers to each field
- Moving the pieces of the `Format` string where they are desired

and

- Associating different `Format` strings with different tabs

As I said, labor intensive but not rocket science.

Now, however, I have a situation where the user can select between one and five sort nodes. Therefore the user can chose any of 120 orders (five possibilities in the first node, four in the second, three in the third, two in the fourth makes 120 possible combinations; this is not just a theoretical maximum, since I don’t have to code for the user not having elements at the second – therefore all remain -- levels).

Thus, to cover all the possibilities, I have to create `Format` strings for each in anticipation. I also have to create a `Case` statement to test for each combination. The phrase "no [expletive deleted] way" comes immediately to mind. My official response, when asked for this feature, was that it would be "an enormous undertaking and I feel it is best made a low priority request." (Two ways of saying the same thing, aren’t they?)

What would be perfect would be to create five strings, one for each variable associated with each possible node. Then, hijack the part of the `Format` string appropriate to that node and prime the variables in INIT:

```
PLUString  = '68L(2)|M~PLU~@s16@#1#'
VNDString  = '39L(2)|M~Vendor~@s8@#7#'
PartString = '69L(2)|M~Part Number~@s16@#6#'
UserString = '37L(12)|M~User Sort~L(2)@s3@#9#'
DeptString = '43L(12)|M~Department~L(2)@n_3@#8#'
```

I would also require strings for the "fixed" fields (fields that always appear):

```
DescString = '120L(2)|M~Description~@s30@#2#'
OldOH      = '49R(4)|M~Old On Hand~L(2)@n-_9.2@#3#'
NewOH      = '52R(9)|M~New On Hand~L(2)@n-_9.2@#4#'
ScanS      = '31C|M~Scan In~L(2)@s1@#5#'
```

The string used to store the sort order is `PIH:Parameters` and I would loop through it, picking up the user’s selections and concatenating a big `Format` string.

I already *do* loop through `PIH:Parameters` to create a string to use in the `Additional Sort Order` prompt (i.e., for `AppendOrder`), so I would piggyback on this existing code:

```
ParmOrder = ''
Loop i# = 1 to Len(Clip(PIH:Parameters))
  Case PIH:Parameters[i#]
  Of 'P'
    ParmOrder = Clip(ParmOrder) & '+PID:PLU,'
    FormatString = Clip(FormatString) & PLUString
  Of 'N'
    ParmOrder = Clip(ParmOrder) & '+PID:PartNumber,'
```

```
        FormatString = Clip(FormatString) & PartString
      Of 'U'
        ParmOrder = Clip(ParmOrder) & '+PID:UserSort,'
        FormatString = Clip(FormatString) & UserString
      Of 'V'
        ParmOrder = Clip(ParmOrder) & '+PID:Vendor,'
        FormatString = Clip(FormatString) & VNDString
      Of 'D'
        ParmOrder = Clip(ParmOrder) & '+PID:Department,'
        FormatString = Clip(FormatString) & DeptString
      End
    End
    ParmOrder = '+PID:Number,' & ParmOrder[1:Len(ParmOrder)-1]
```

(The last line removes the final, hanging, comma from the string passed to `AppendOrder`.)

Then, in the sheet's `NewSelection` embed, a simple

```
    Of 6
      ?Browse:1{Prop:Format} = FormatString
```

would do it.

Unfortunately, the documentation on `Prop:Format` is quite explicit: `Prop:Format` takes "A string constant specifying the display format." `Prop:Format` takes a string constant, not a variable. SoftVelocity is quite good about documenting when a constant or equate or variable is acceptable. Only "constant" is referenced here. I am inclined to believe them.

Even more unfortunately, I had already put all the code in place before I checked the docs. So I decided to try compiling the code.

The worst that could happen is that the compiler would choke on it. It didn't.

So now the worse that could happen is that it would GPF at runtime. It didn't.

Having averted two worst case scenarios, the next worse thing is that the code wouldn't work. But, at least, it wouldn't GPF.

It did (work, that is).

I think we have a certified "documentation bug." I suppose Figure 7, above, gave it away, didn't it?

Summary

I’m not quite sure what I learned in this exercise and I write articles to document my learning. I already knew I could parse the `Format` string created by the screen painter. I already knew that I had to add the queue numbers if I wanted to rearrange queue elements in a browse. I did not know that `Prop:Format` would take a variable (though I did so assume).

I guess I discovered that the documentation is wrong. But I would certainly never advise ignoring the documentation. So there is no lesson in this.

So, this is what I want to pass on to you: master the first two steps of "How to Display the Sort Field First on a Multi-Key Browse." Once you have done that, you can prime all manner of variables and create new browse formats on a mouse click’s notice to your heart’s content.

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

Reader Comments

Add a comment

Clarion Magazine

A Better DATE Function

by Carl Barnes

Published 2006-01-09

Here's a replacement for the Clarion `DATE` function that fixes some of `DATE`'s limitations.

I frequently want to go back *x* days or months. `DATE()` cannot handle something like `DATE(1,-15,2000)`. This function also handles going forward before calling `DATE()` so the C5 leap year problem is fixed with `DATE(14,29,1999)`.

Here's the source, or you can use the version in the downloadable APP, below.

```
DateFixed2  PROCEDURE(long _Month,long _Day, long _Year)

RetDate     long,auto
YrsAdj      long,auto

    CODE

    ! Passing Date Function a Negative or Zero,
    ! Month or Day Does not work, it returns
    ! a bad value. There was a bug with 14/29/1999
    ! not seeing  it as 2/2000 and as a Leap Year
    ! So try to get the date parts into correct range
    ! values before using Clarion Date()

     !If all zeros then do not try to adjust and get confused
```

```
IF ~_Month AND ~_Day AND ~_Year THEN RETURN 0.

! Date cannot deal with Negative or zero Month
! so how many years are we off + 1
if _Month < 1
   ! advance Month forward
   YrsAdj = 1 + ABS(_Month) / 12
   _Month += YrsAdj * 12
   ! adjust years back
   _Year  -= YrsAdj
end
! If year passed as YY for 2000 (i.e. 00) and
! adjusted with -1 this fixes -1 to be 1999
if _Year < 0 then _Year += 2000.

! In Leap Years but with Months > 12
! as noted by Clarion Mag
if _Month > 12
    YrsAdj = (_Month - 1) / 12
   _Month -= 12* YrsAdj
   _Year  += YrsAdj
end

! In case 99 gets passed and +1 to
! 100 else (14,29,99) fails
if _Year > 99 and _Year <= 999 then
   _Year += 1900
end

! Date() cannot deal with Negative Day
if _Day < 1
   RetDate = date(_Month, 1, _Year) - 1 + _Day
else
   RetDate = date(_Month, _Day, _Year)
end

return RetDate
```

Download the source

Carl Barnes is an independent consultant working in the Chicago area. He has been using

Clarion since 1990, is a member of Team TopSpeed and a TopSpeed Certified Support Professional. He is the author of the Clarion utilities CW Assistant and Clarion Source Search.

Reader Comments

Add a comment

- » I'm a little confused... What does DATE(1,-15,2000) used...
- » Paul, You are correct that offsetting a Clarion date by a...
- » Sorry guys, my fault. I somehow munged the HTML and got an...

Clarion Magazine

The ClarionMag Blog

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

XML All blog entries
XML All new items, including blogs

Blog Categories

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

Ugly website. Brilliant content.

Direct link

Posted Wednesday, January 25, 2006 by Dave Harms

That's the motto of incompetech.com, a terrific site which has several things I just happened to need today. One was a sheet of paper showing a grid of dots. I thought of how

I could do this myself, and then realized that grid paper must be available somewhere on the web. Google led me to the online graph/grid paper generator at incompetech.com. You input your specifications, and you get a PDF with your graph/grid. Too cool. I also printed a perfectly usable calendar. Also on this site: loops and royalty-free instrumental music, not to mention The Gallimaufry.

Update: Coincidentally, Mark Coffey at Cool Tools picked up on the incompetech site, but for a different purpose:

> OK, so I wanted to sit down and workout a grand plan for my new garden, so I figure a pencil and some graph paper is the way forward.
>
> Just finding some simple 2mm graph paper with 1cm semi bold and 2 cm bold turned out to be a near impossible task. Then I discovered the Graph Paper PDF Generator at incompetech.com .
>
> It does plain paper, lined paper, multi width, hexagonal, even semi-bisected trapezoid! All completely customizable. And it's free!

Crossposted at Knobblegrud.

Gary James on the loose

Direct link

Posted Friday, January 20, 2006 by Dave Harms

Gary James has parted (amicably) with CapeSoft, and is striking out on his own. You can keep up with Gary at his blog. From his departure announcement:

> For the past 7 years I have been working on StrategyOnline after hours and on weekends, and I'm now going to be putting all my energy into it to see if I can go it alone. For numerous reasons the timing is just right. I will still be working mostly in Clarion, and will still be "around" on the newsgroups, MSN, Skype, DevCons etc, so if you were one of my "CapeSoft clients" please keep in touch, I'd love to hear from you. The Clarion community is something really special, and I have loved building accessories for you. I

> will still be building accessories (among other things), only they will now be branded as “StrategyOnline”.

Gary's done some great work for Capesoft; I'm sure he'll be equally successful on his own, and will eventually get that BMW!

Conversion program gets thumbs down (updated to thumbs up!)

Direct link

Posted Friday, January 20, 2006 by Dave Harms

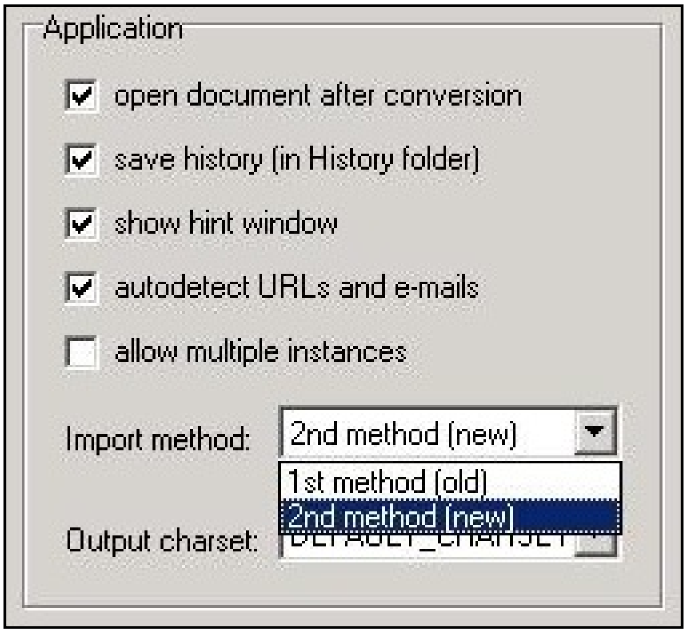
Yesterday I posted a grab bag of links that have been gathering dust on my computer, including the ABC Amber Clarion Converter. I heard back from one developer who is less than impressed:

> Well, this semi-commercial program so far is more disappointing than the freebie converters.
>
> Some simple non-memo field .DAT files cause the program just to shut down without doing a thing, while others do open and the memo field ones never come close to opeming. Files that the freebie converters work on fail in this software.

I haven't seen this developer's data or tried the program myself, and it's possible there's something excessively funky about the data. But the converter certainly doesn't look promising.

Update

Looks like I jumped the gun on that report - the ABC Amber Clarion Converter does in fact work, but you may have to use the 2nd method (new), as shown in the image at right. From the developer testing the conversion:



Application

☑ open document after conversion

☑ save history (in History folder)

☑ show hint window

☑ autodetect URLs and e-mails

☐ allow multiple instances

Import method: 2nd method (new)

Output charset: DEFAULT_CHARSET

> After that reset, I attempted to open a DAT file that I couldn't open before and that I knew had a Memo field in it. Success !!! The Clarion Converter read it successfully and showed me the reference to the Memo field.
>
> Then I decided to export the records to a Word .doc file to see whether it still recognized the Memo field in the process. It did its conversion thing and automatically opened up Word 2000 on the PC I was on and showrf the newly converted data right in front of me. Then I decided to export it to the ubiquitous .dbf file format. This is where the file conversion fell down, but only because the DAT file included a VCL field type 1B that the converter said it couldn't write to a .dbf. The .dbf file format doesn't support that Clarion field type, even though the Converter was able to successfully convert and present it in the Word .doc file. The solution to get any data at all from that .DAT file into a .dbf file format is to exclude that field type from the conversion process. I excluded that one field and the conversion processs from a Clarion 2.1 .DAT file successfully went over into a .dbf file format even with the Memo field and its contents. Very nice.

Dunno what a VCL field is, but it's nice to hear that everything else is working.

File Manager 3 mini-review

Direct link

Posted Thursday, January 19, 2006 by Dave Harms

Stuart Andrews at DevDawn has a short review of CapeSoft's File Manager 3:

> Basically, it's like this. In Clarion you have a dictionary, which contains the file structures, and you have an app, which contains the windows, the code, the look. When you change a field within a table, the physical file is required to undergo an upgrade. The process to upgrade your user's files was complicated (as opposed to getting them to send you the file, upgrading yourself through the dictionary interface, and then sending it back), and created a lot of hassle.
>
> FM3 takes all this hassle, crumples it up, throws it into the ditch, because the bin is too good for it, fills the ditch with gasoline, lights the ditch, then laughs at the stupidity of it all.

Grab bag of links and tips

Direct link

Posted Tuesday, January 17, 2006 by Dave Harms

The back room here at the ClarionMag blog is starting to look like Fibber McGee's closet. Time to clean up:

- Here's a massive list of .NET, ASP.NET, and SQL links. [HT: Mike Pickus]
- Get paid/pay for knowledge. From the Lockergnome Helpshare web site: "The HelpShare technology enables people to ask questions, receive answers from other Web users and financially reward the most useful answers. By adding email functionality (alerting experts when your question is posted), credibility ratings (to judge the credibility of the answerers) and e-commerce functionality (to incent the best users to answer tough questions) to the time-tested newsgroup model, HelpShare provides an interactive, easily accessible, and efficient problem-solving tool."
- Wayne Price points out Fiddler, a nifty HTTP debugging proxy that lets you inspect, and modify, HTTP traffic. Looks pretty handy for anyone involved with web services, SOAP, etc.
- If you've done any Windows API work, you've probably used callback functions. In .NET, callbacks are type safe and are called delegates, and will be an important part of Clarion.NET. This article from MSDN Magazine

covers the basics.

- You can get a .NET/ASP.NET-related news feed at DotNetSlackers.com, says Mike Pickus.
- Code escrow comes up from time to time in the newsgroups. A while back Chris Peacock pointed out CodeKeep.
- Writely is a web-based word processor. Just in case you're interested. [HT: Bill Kinnon]
- If you do Java work, note that a bunch of Sun's Java tools are now free. I do some server-side Java work, but so far I'm happy with Eclipse.
- Ben Brady stumbled across a link to a Clarion file converter.
- Skype 2.0 has video conferencing, but if you're not happy with Skype's effort you could have a look at Festoon. I haven't used either.
- If you use MySQL's InnoDB file system for transaction-oriented files, note that back in October Oracle acquired InnoBase. Hey, you think that will have a future impact on MySQL's ability to provide transaction-oriented processing?

Okay, I can see the back wall again!

Andrew's Kitchen photos updated

Direct link

Posted Tuesday, January 17, 2006 by Dave Harms

Leroy Schulz has updated the "Best of Andrew's Kitchen 2005" photos with new entries from Rex Kersley, Tony Tetley, Shawn Simmons, Andrew Guidroz, and Mark Riffey. Get your entries in now. No pictures of food yet, unless you count the snow cones - Andrew's Kitchen is a Clarion Magazine newsgroup hosted by his Cajunness, Mr. Guidroz, and although food gets talked about often enough in AK, it's really just the place ClarionMag folks hang out. Like the kitchen. If you're a subscriber and you don't yet have access, let me know. If you're not a subscriber, there's no time like the present, particularly with Clarion.NET and C7 going into pre-release this month. Subscribe now.

Clean that disk

Direct link

Posted Monday, January 16, 2006 by Dave Harms

Here are a pair of disk cleaning programs you might want to try out. Geoff Bomford, while stopping short of unconditional recommendation, has had a good experience so far with CCleaner (Crap Cleaner), a freeware, wizard-style, automated system optimization and privacy tool. Brett Carnaby likes SpaceMonger, which is also one of my favorites. SpaceMonger is a manual, visual tool, and a great way to see how much space is used by what. I do find it a bit prone to crashing, however.

The shirtless, shoeless Clarion salesman

Direct link

Posted Thursday, January 12, 2006 by Dave Harms

There's an interesting thread started by Bob Sutton in softvelocity.clarion.chat, titled Clarion - 6 degrees of separation. Bob asks, "I have been
wondering just how everyone found out about Clarion?"

Dave Schwartz replies:

> Okay, I think I would win "the strangest tale" award.
>
> The year was 1992.
>
> I had a guy walk into my office one day wearing shorts, no shirt and no shoes. He said he was down on his luck.
>
> He said that he was on the way from Sacramento to Pompano Beach where he had
> gotten a job with Clarion. The night before he and his girlfriend had an argument and when he woke up that morning everything he owned and all of his
> money was gone.

> He talked me into giving him about 15 minutes to demo this unique programming language called Clarion Professional Developer. So, having no
> expectation of anything, I gave him the time and was very impressed.
>
> He said he would leave me with the manual (it was in a 3-ring binder) and the software if I'd just buy him a bus ticket to Pompano.
>
> I bought the bus ticket, gave him $40 for food on the way, and dropped him at the bus station.
>
> With Clarion and me it was love at first sight. I went to my local Walden Software (remember them?) and bought a legal copy (for $549 I think).
>
> Oh, and as I recall, we gave him a shirt and some shoes.
>
> Regards,
> Dave Schwartz

Yes, Dave, you win.

But who was the guy?

Laying out controls in the new Clarion IDE

Direct link

Posted Wednesday, January 11, 2006 by Dave Harms

Okay, I'm teasing you a little on this one. I don't have a preview of the new Clarion IDE per se, and I'm not exactly sure what will be in the pre-beta releases expected this month. But as I've been doing a bit of Visual Studio C# coding lately, I thought I'd take a look at a beta release of SharpDevelop 2.0, since C7 and Clarion.NET, in gold release, will at least in part use the SharpDevelop 2.0 user interface code.

The folks at SharpDevelop follow the VS user interface model pretty closely. It's a stretch to call SharpDevelop a Visual Studio clone, but certainly it's easy to move between the two environments.

Specifically, I wanted to look at the WinForms designer. In Clarion versions to date, laying out controls on a form is still a pain, although you can make it bearable with some intelligent grid size decisions. In Visual Studio, moving and sizing controls is, pun intended, a snap - the forms designer intelligently snaps controls relative to the window, and other controls, as you move and size, making design a breeze. I don't know if that capability is in SharpDevelop 1.0, but it's certainly there in the 2.0 beta, and I'm really looking forward to seeing intelligent control layout in Clarion.NET.

Whether C7 enjoys the same window formatter niceties is another issue - as Bob Z noted in the June 30th podcast (read the transcript), the Win32 formatter and the .NET formatter are two very different beasts.

C7, Clarion.NET early releases still on track for January

Direct link

Posted Monday, January 09, 2006 by Dave Harms

Early releases of C7 components and Clarion.NET are still on track for January, as noted in this blog entry by Bob Z. The C7 pieces include the runtime library and compiler, with new base controls that fully support XP themes. If you're part of the pre-beta program, I imagine you'll be able to run the compiler against your existing, generated (or hand-coded) source.

As previously reported, a significant part of the new IDE is code licensed from the makers of SharpDevelop. The good news is that the initial C7 Win32 IDE release, and the final integrated C7/Clarion.NET IDE, will use the SharpDevelop 2.0 code base. The initial release of the Clarion.NET IDE will use the 1.0 base, but will be ported soon. The SharpDevelop 2.0 beta was only released Dec 19.

Again, those of you in the pre-beta program can expect to see significant IDE changes as time goes on, so don't get too hung up the detailed IDE appearance or functionality of the first release of either C7 or Clarion.NET. If you were part of the Clarion for Windows alpha process, you'll know what I mean. Ross Santos and I began writing Developing Clarion for Windows Applications during the alpha, and halfway through the book we had to completely reorganize the content because of major (and highly beneficial) changes made by the development team.

And yes, as Bob says, there will definitely be coverage of the early releases in Clarion Magazine!

[Last 25 entries] [Last 50 entries] [All entries]