

# Clarion Magazine

## Clarion News

---

- » [Clarion Roadmap Updated](#)
- » [C7 Editor Videos](#)
- » [Free icons, Christmas-Themed Wallpaper](#)
- » [SetupBuilder 5 Developer Edition End of Sale Notice!](#)
- » [SetupBuilder 5 Professional Edition End of Sale Notice!](#)
- » [SetupBuilder 6.0 Professional Edition](#)
- » [Clarion FreeImage Project 3.9.2.1](#)
- » [SetupBuilder 6.0 Developer Edition Released](#)
- » [Smart-Type 1.1](#)
- » [Clarion Mappoint Templates 2.0.259](#)
- » [WindowID v1.05](#)
- » [StrategyOnline Holiday Schedule](#)
- » [J-Skype and Skype 3](#)
- » [Dr. Watson Analyser 1.3.0](#)
- » [SQL Programming Series Course Special Offer](#)
- » [iQ-XML 1.23 For 9055](#)
- » [Huenuleufu Products 9055 Compatible](#)
- » [ClassViewer Update](#)
- » [Clarion6 Build 9055 Change Log](#)
- » [Clarion Desktop 2.40](#)
- » [Oak Park COM Domain Special](#)
- » [Clarion Desktop 2.35](#)
- » [xNotes Browse-Form 1.2](#)
- » [WinEvent 1000th Copy Update](#)
- » [Smart-Type Template](#)

- » [Qview Debugging Tool](#)
- » [Ingasoftplus Christmas Offer](#)
- » [CoolFrames 1.02](#)
- » [ClarionTools View Wizard 6.03](#)
- » [Organize 365 Source Code Offer](#)
- » [Clarion Desktop 2.31](#)
- » [Clarion Desktop 2.20](#)
- » [PDF-XChange/Tools SDKs For Clarion V3.601091](#)
- » [Icetips Previewer 2.401](#)
- » [2007 Aussie DevCon Pre-Attendance Survey](#)
- » [Clarion Mappoint Templates 2.0.257](#)
- » [Oak Park Virtual Dedicated Server Sale](#)
- » [xNotes Browse-Form 1.0](#)
- » [FullRecord 1.71](#)
- » [NeatMessage 2.00](#)
- » [CoolFrames 1.01](#)
- » [Create RSS Feeds With xFiles](#)
- » [WinEvent Nears 1000 Users](#)

[\[More news\]](#)

[\[More Clarion 101\]](#)

## Latest Free Content

- » [Source Code Library Goes Gold!](#)

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

## Clarion Sites

---

## Clarion Blogs

---

“ One of my most valuable tools. ”  
▶ Shawn Simmons



## Latest Subscriber Content

---

### [Manifests for Hand Coded Apps](#)

Clarion 6 includes support for XP manifests in APPs. But what about hand coded programs? Mark Riffey shows how to add a manifest to your PRJ.

Posted Friday, December 29, 2006

### [Metadata Management: The Way to a Well-Ordered Enterprise](#)

Metadata is simply data about data. Companies everywhere use metadata, most commonly in their financial systems. But IT has, or should have, metadata as well. Mike Gorman explains how to approach metadata management.

Posted Friday, December 29, 2006

### [Printing a "No Records" Report](#)

The standard behavior for a Clarion, ABC, report, if there is no data, is to pop up a message which says "No records". Suppressing this message is easy, but what if you want to print a report that says "No records"? Bruce Johnson shows how it's done.

Posted Friday, December 22, 2006

### [A Callback Technique for Capesoft's FileManager2](#)

CapeSoft's FileManager2, and its more recent incarnation FileManager3, are amazing products: they completely automate the process of upgrading client databases, whether local or across a network, and make it easy to do file maintenance. Randy Rogers shows how to add a callback procedure to FM2 to provide user feedback during startup.

Posted Thursday, December 21, 2006

### [No! You Can't Cancel](#)

Clarion developers are a nasty bunch. They do not want to let their end users cancel forms. At least it seems so from the frequency with which questions are asked about trapping the Escape key or Windows' red X. Dr. Parker shows how it's done.

Posted Wednesday, December 13, 2006

### [A Customized Deep Assign Function](#)

The deep assign operator (:=) is a powerful tool, but it doesn't work in all circumstances. Alan Telford shows how to extend deep assign functionality using WHO and WHAT.

Posted Tuesday, December 12, 2006

### [How To Run Multiple Instances of Clarion \(video\)](#)

Richard Rose's Camtasia presentation on how to run multiple instances of Clarion, by special arrangement with the UK Clarion User's Group. This is a 14 minute, 63 meg AVI file. Includes registry files and batch file. Please note that although you can have multiple instances open, you can only have one instance compiling/generating at one time. If you have trouble downloading please delete any partial downloads and do a Save As.

Posted Tuesday, December 12, 2006

### [Source Code Library Goes Gold!](#)

The Clarion Source Code Library has officially gone gold! The release version is 2006.11.30 and includes all the November source - if you downloaded in the last couple of days you have this version.

Posted Wednesday, December 06, 2006

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

## Source Code

---

### [The ClarionMag Source Code Library](#)

Clarion Magazine is more than just a great place to learn about Clarion development techniques, it's also home to a massive collection of Clarion source code. Clarion subscribers already know this, but now we've made it easier for subscribers and non-subscribers alike to find the code they need.

The Clarion Magazine Source Library is a single point download of all article source code, complete with an article cross-reference.

[More info](#) • [Subscribe now](#)

## Printed Books & E-Books

---

### [E-Books](#)

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

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

## [Printed Books](#)

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



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

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

## **From The Publisher**

---

### [About Clarion Magazine](#)

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

### [Subscriptions](#)

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

### [Satisfaction Guaranteed](#)

For just pennies per day you can have this wealth of Clarion development information at your fingertips. Your Clarion magazine subscription will more than [pay for itself](#) - you

have my personal guarantee.

Dave Harms

## ISSN

---

### [Clarion Magazine's ISSN](#)

Clarion Magazine's [International Standard Serial Number](#) (ISSN) is 1718-9942.

### [About ISSN](#)

The ISSN is the standardized international code which allows the identification of any serial publication, including electronic serials, independently of its country of publication, of its language or alphabet, of its frequency, medium, etc.

# Clarion Magazine

## Clarion News

[Search the news archive](#)

### [Smart-Type Version 2](#)

Smart-Type Version 2 is now available. This release is a complete rewrite from pure template code to an ABC compliant class with an optional template wrapper. You can still add it to your applications using the same template system - the template looks very similar. If you are upgrading, you will need to re-fill in the template prompts for each control though, as Smart-Type now takes the field equate instead of the use variable as a parameter. You can also add it by hand-code in three easy steps, and it works in multi-DLL applications. Smart-Type comes with a template and a hand coded example.

Upgrades are free.

Posted Tuesday, January 02, 2007

### [BackItUp 1.4.0](#)

BackItUp 1.4.0 is now available. This release includes a fix for screen flickering during backup.

Posted Tuesday, January 02, 2007

### [Code Commentor 2.0.3](#)

A time token, %TM% has been added to the Code Commentor.

Posted Tuesday, January 02, 2007

### [AFE v2905 Server Upgrade Released](#)

AFE v2905 Server is now available. This release includes: Latest FaxMan4 components;

Fax reception; Completely rewritten setup APP in C5.5, C6 & C6.1+; Rudimentary fax viewer APP in C5.5, C6 & C6.1+; Rewritten user HTML help; Expanded developer notes; Completely rewritten server installer; Removal of printer driver requirement, and much more. If you purchased or upgraded AFE within the last two years you should have already received an email with information to get your free upgrade to v2905. If you feel you are in this group and have not received your notice send an email to Products (at) LodestarSoftware (dot) com

Posted Tuesday, January 02, 2007

### **[Code Commentor 2.0](#)**

Version 2.0 of the Code Commentor is now available. Major changes for this version include: For the contract programmer, the system now supports multiple companies; Task history still defaults to 10 entries but is now user configurable from the File-System Configuration menu for as many entries as you would like; Data storage has been moved from the registry to data files stored in the program directory.

Posted Tuesday, January 02, 2007

### **[Abspoel.nl Clarion Site](#)**

The Abspoel Clarion site has been updated to the newest version of Wordpress and a new theme.

Posted Tuesday, January 02, 2007

### **[Clarion Roadmap Updated](#)**

Bob Zaunere has released an update to the Clarion roadmap. C7 is expected to go to third party vendors and other beta testers in late January/early February. Look for sneak peeks in ClarionMag in advance of the release. Clarion.NET is on track for release shortly after C7, and the Compact Framework designer has been completed ahead of schedule (initially planned for the next release of Clarion.NET).

Posted Saturday, December 30, 2006

### **[C7 Editor Videos](#)**

Bob Foreman has posted a number of flash movies demonstrating the C7 editor. Features discussed include code folding, the structure designer (including a look at Report structures), search and replace, and editor configuration options.

Posted Friday, December 22, 2006



## **[Free icons, Christmas-Themed Wallpaper](#)**

Icons-Icons is giving away a nice little selection of free icons for December, along with a Christmas-themed "sale" graphic, and a pretty Christmas-themed wallpaper.

Posted Thursday, December 21, 2006

## **[SetupBuilder 5 Developer Edition End of Sale Notice!](#)**

SetupBuilder 5 Developer Edition comes to End of Sale (EOS) on December 20, 2006. SetupBuilder 5 Professional will still be supported by Lindersoft until March 31, 2007. Service packs and hotfixes will be released up until this date, if determined by Lindersoft to be necessary. Lindersoft strongly recommends customers upgrade to the latest version of SetupBuilder 6 Professional or Developer Edition as soon as possible to maintain the highest level of support, performance and reliability. If you have a current SetupBuilder Developer Edition maintenance subscription, you will be entitled to receive all minor and major upgrades for your product released during the term of your subscription free of charge. If your maintenance subscription expired, you can reactivate it. You will receive a new subscription which will allow you to download and use an additional 12 month's worth of releases.

Posted Thursday, December 21, 2006

## **[SetupBuilder 5 Professional Edition End of Sale Notice!](#)**

SetupBuilder 5 Professional Edition comes to End of Sale (EOS) on December 20, 2006. SetupBuilder 5 Professional will still be supported by Lindersoft until March 31, 2007. Service packs and hotfixes will be released up until this date, if determined by Lindersoft to be necessary. Lindersoft strongly recommends customers upgrade to the latest version of SetupBuilder 6 Professional or Developer Edition as soon as possible to maintain the highest level of support, performance and reliability. If you have a current SetupBuilder Professional Edition maintenance subscription, you will be entitled to receive all minor and major upgrades for your product released during the term of your subscription free of charge. If your maintenance subscription expired, you can reactivate it. You will receive a new subscription which will allow you to download and use an additional 12 month's worth of releases.

Posted Thursday, December 21, 2006

## **[SetupBuilder 6.0 Professional Edition](#)**

Lindersoft has released SetupBuilder 6.0 Professional Edition. Price starts at \$179 USD

for a royalty-free usage license. A trial version is available. To install the full (licensed) version 6.0, please use your current password, serial number and Subscription Plan ID. SetupBuilder 5.6 and SetupBuilder 6.0 can coexist on the same machine.

Posted Wednesday, December 20, 2006

### **[Clarion FreeImage Project 3.9.2.1](#)**

This release of Clarion FreeImage Project adds basic support for the new functions in the library. The classes now contain methods to implement the FreeImage composite function. This allows you to alpha blend an image onto a background color or another image. There's also a fix for a memory leak that occurs if you destroy the Clarion FreeImage object before the child window. An update to the Smooth Scroll Control is also available on the related projects page.

Posted Wednesday, December 20, 2006

### **[SetupBuilder 6.0 Developer Edition Released](#)**

Lindersoft has released SetupBuilder 6.0 Developer Edition. SetupBuilder(R) Version 6 is a powerful and easy to use Installation Authoring and Configuration Management environment for developers and organizations deploying applications to the Microsoft(R) Windows(R) Platform, including Windows x64 and Windows Vista(R). Developer Edition gets you ready for Windows Vista and .NET application deployment (Clarion.net, C#, etc.). Lindersoft's iDeploy Internet distribution technology allows you to very easily deploy and update your applications over the Internet. The powerful built-in UpdatePROTECT feature provides a way to handle software subscriptions. This technology is well suited for software subscription services in which the end-users subscribe, for instance on a monthly, quarterly, semi-annually, etc. basis. SetupBuilder 6 Developer Edition starts at \$299 USD for a royalty-free usage license. A trial version is available.

Posted Monday, December 18, 2006

### **[Smart-Type 1.1](#)**

Smart-Type 1.1 is now available. This release includes improved support for VSCROLL on text controls and learn-while-you-type mode.

Posted Monday, December 18, 2006

### **[Clarion Mappoint Templates 2.0.259](#)**

WC Software Development Inc. has released Clarion for Mappoint Templates 2.0.259. Changes include several new methods, C6 9055 compatibility, FAQ section, and template changes.

Posted Monday, December 18, 2006

### **WindowID v1.05**

WindowID v1.05 is now available. Fixes problem with no WindowID button in a browse when toolbar buttons deleted. This release is compatible with Clarion Desktop.

Posted Monday, December 18, 2006

### **StrategyOnline Holiday Schedule**

Tomorrow evening Gary James is closing shop for just over a week. He'll be checking email and orders during that time. Look for new versions of J-Skype, Organize365 and Clarion Desktop within the next 24 hours.

Posted Thursday, December 14, 2006

### **J-Skype and Skype 3**

J-Skype has been tested with Skype version 3, and it seems to work fine, but note that after you upgrade to Skype 3 you must reboot before the new Skype API will work correctly. And the price increase from beta is imminent...

Posted Thursday, December 14, 2006

### **Dr. Watson Analyser 1.3.0**

A new version of the Dr. Watson Analyser utility is now available for download.

Posted Thursday, December 14, 2006

### **SQL Programming Series Course Special Offer**

SoftVelocity's SQL programming course has detailed coverage for MS SQL, SQL Anywhere, Oracle, Firebird, PostgreSQL, MySQL, Pervasive SQL, and using ODBC to connect to any backend. Topics covered include: Best design practices for SQL applications; Using driver strings; Direct queries via PROP:SQL; The power of TurboSQL; Using stored procedures; Using server side triggers; Optimizing keys and indexes, and more. Save \$50 until Jan 15.

Posted Wednesday, December 13, 2006

### **[iQ-XML 1.23 For 9055](#)**

iQ-XML 1.23 is now available. This release has been compiled for C6 build 9055.

Includes a time zone stamp fix and an increase in the write size limitation.

Posted Wednesday, December 13, 2006

### **[Huenulefu Products 9055 Compatible](#)**

All Huenulefu products have been tested with Clarion 6.3 9055 and work fine. Since these are all 100% source code all you have to do is recompile your app with 9055 when you get it.

Posted Wednesday, December 13, 2006

### **[ClassViewer Update](#)**

An update to the Keystone ClassViewer program is now available for download. This update corrects a couple of parsing errors, has increased buffer sizes to handle longer ini file entries, and adds support for the undocumented [redirection macros] section in the c60ee.ini file.Regards

Posted Tuesday, December 12, 2006

### **[Clarion6 Build 9055 Change Log](#)**

The change log for C6 9055 is now up. Changes include bug fixes and some SQL enhancements. Also only one parent structure is allowed in the declaration of a GROUP, QUEUE, CLASS or INTERFACE. Previously additional parent structures were allowed but ignored.

Posted Tuesday, December 12, 2006

### **[Clarion Desktop 2.40](#)**

Clarion Desktop 2.40 adds a "wide-screen" mode and includes an important bug fix.

Posted Tuesday, December 12, 2006

### **[Oak Park COM Domain Special](#)**

By request the Oak Park .COM domain special is back. Register your .COM domain for %6.95 per year (new and transfers). Each registration or transfer includes over 100 dollars worth of extras.

Posted Tuesday, December 12, 2006

### **Clarion Desktop 2.35**

New features include: View headline dates in the various news sections; Set the date format: Internal optimizations and cosmetic improvements.

Posted Tuesday, December 12, 2006

### **xNotes Browse-Form 1.2**

New in xNotes Browse-Form 1.2: Bug fix for icons under Clarion 5.5; Important changes to xNotesBFCreateNote template; Updated demo: Minor template changes.

Posted Tuesday, December 12, 2006

### **WinEvent 1000th Copy Update**

Current 969 copies of WinEvent have been sold. The purchaser of the 1000th copy of WinEvent will receive their copy for free, plus any other CapeSoft product they purchased at the same time also for free.

Posted Tuesday, December 12, 2006

# Clarion Magazine

## Manifests for Hand Coded Apps

by Mark Riffey

Published 2006-12-29

I've never been much of a hand-coder, well, at least not since leaving the mainframe world. It's not that I can't do it, I just haven't seen a need to with the projects I've worked on. Sure, there have been tiny little exceptions, but no full-blown hand-coded apps.

Then one fine Montana afternoon I was sitting in a coffee shop with a hand-coder I first met at ETC 2002. Among other things, I was showing him an AppGen creation that I was working on for him. He noticed that my buttons were a lot purtier (that's a highly technical [term](#)) than the buttons in his hand-coded apps. My app looked nicer because Clarion 6 now has template support for Windows XP Themes, a way of applying a consistent look and feel to Windows controls.

He wondered aloud how to make this happen for his hand-coded app. I said I wasn't sure, but I thought it was a matter of adding a compiler pragma and a manifest XML file. I mentioned that I could ask a few guys that I know who do a fair bit of hand coding - as I was sure they had probably done this - and get back to him.

The task turned out to be even easier than I thought.

### Manifest destiny

A manifest is a XML file that tells Windows XP (etc) how to "pimp my app" and make it look like all the other slick-looking apps out there. A manifest won't change how the application looks on earlier versions of Windows.

**Definition from my unofficial ClarionMag slang dictionary:** "Pimp my app" - A reference to the MTV car customization show, "Pimp My Ride": Take steps to improve the cosmetic appearance of a standard Windows application by using an XML visual control parameters file to automatically display visual control enhancements that are available to applications when running on Windows XP and subsequent versions of Windows

So how to apply a manifest to a hand coded app? Later that weekend, I thought I'd look in the Clarion docs. It turns out that there's a short blurb about the use of the manifest statement in the .exp file, indicating that the manifest needs to be referenced in the "Library, object and resource" part of the project.

I now knew *where* to reference my manifest file, but I didn't *have* a manifest file. I looked on my hard drive (with Google Desktop) for the filespec \*.exe.manifest file, but didn't have a single one. Apparently the Clarion IDE creates one long enough to run the linker and then cleans up after itself.

Googling beyond my desktop, I found a standard manifest file on a Visual Basic page, and modified it slightly:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity
  version="1.0.0.0"
  processorArchitecture="X86"
  name="Microsoft.ClarionMagStyles"
  type="win32"
/>
<description>Clarion Magazine Manifest for HandCoded Apps</description>
<dependency>
  <dependentAssembly>
    <assemblyIdentity
      type="win32"
      name="Microsoft.Windows.Common-Controls"
      version="6.0.0.0"
      processorArchitecture="X86"
      publicKeyToken="6595b64144ccf1df"
      language="*"
    />
  </dependentAssembly>
</dependency>
</assembly>
```

Russ Eggen later pointed out to me that ABCHAIN.TPL contains pretty much the same XML code, starting at line 866. There is a minor difference in the description:

```
<description>Clarion 7 application.</description>
```

Paste the XML into Windows Notepad or your favorite text editor. Save it into your development folder as appname.exe.manifest where appname is the name of your application. This is a requirement. REPEAT: The manifest file *must* be named after your application EXE filename and live in the folder where your app is when it is linked.

There is *no* need to change any of the stuff in the XML source, though my tinkering has shown that the <description> tag and the name attribute on the <assemblyIdentity> tag can be changed. However, such changes are pointless. No one ever sees these files but you.

To test the manifest, simply ensure that it's in your application directory as noted above, and add it to your project. Actually the manifest doesn't *have* to be added to your project, but if you don't do this, you'll need to ship the manifest file with your app. So add it to the project.

Figure 1 shows the example application without the manifest, and Figure 2 shows the app with the manifest.

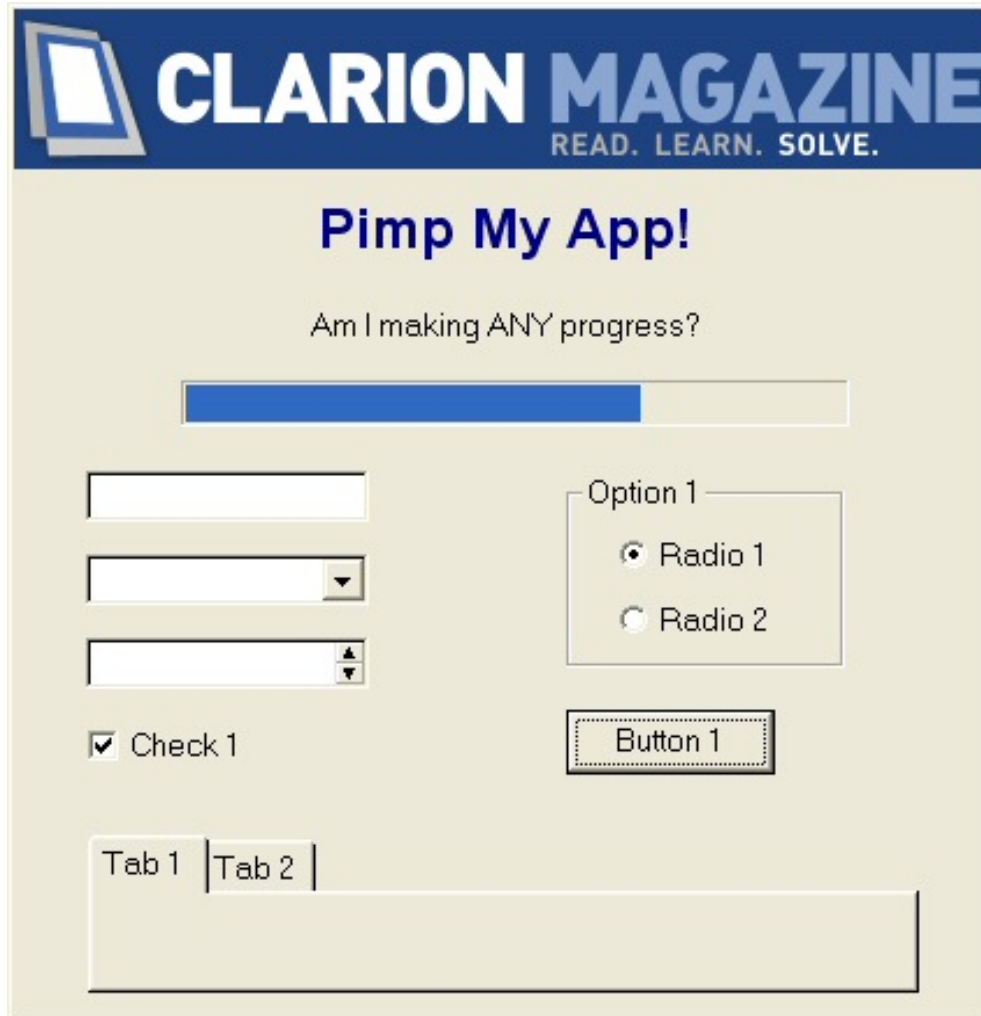
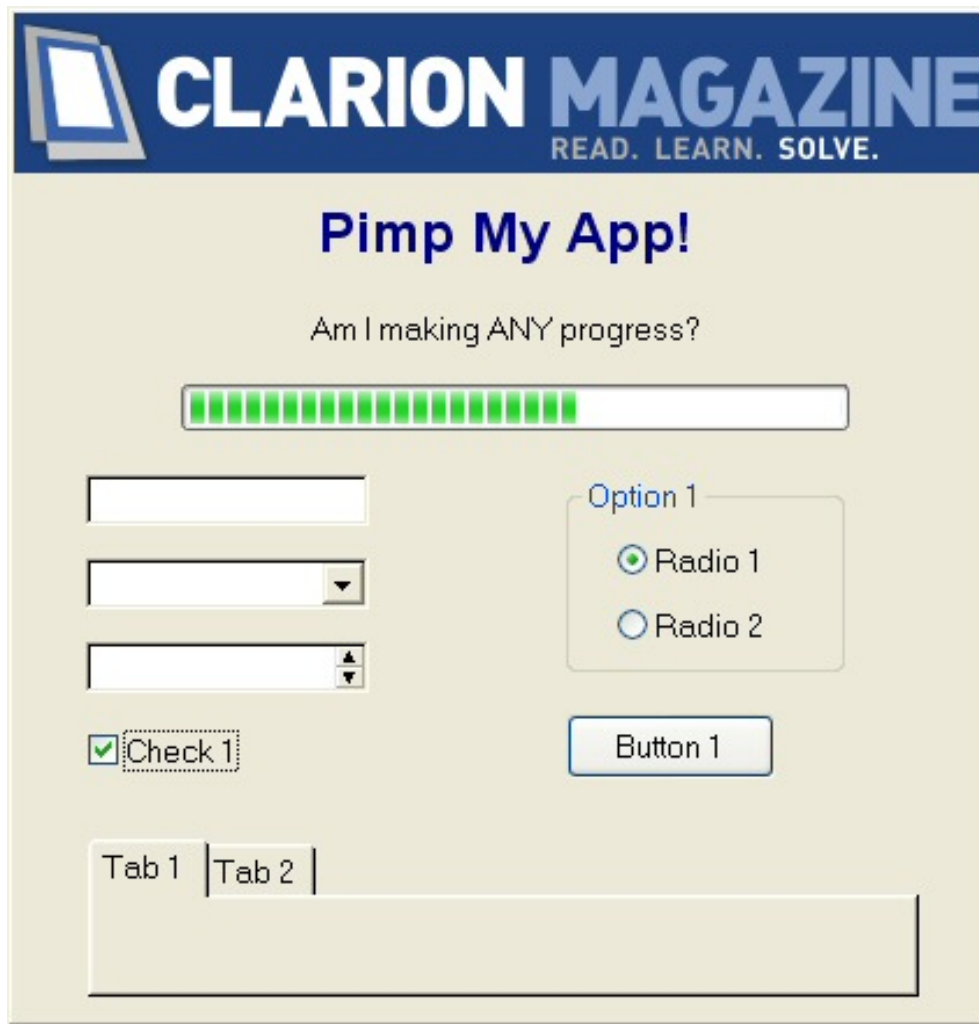


Figure 1. C6 application without manifest





**Figure 2. C6 application with manifest**

Recompile and run. That's it. You don't need to make any .exp file changes, nor changes to any code.

Note that Windows XP's manifest support doesn't know what to do with the non-standard Windows controls that Clarion uses. These are called "owner-drawn" controls. The result of this is that not all controls in your app will be XP-ized and that might not produce a pleasing mix of appearances - that's your call. An owner-drawn control is a Windows control that doesn't come from the standard Windows library - instead it is drawn by the application, or in the case of Clarion applications, by the clarion runtime library. One example is the Sheet/Tab control that Clarion provides. According to the [Clarion Roadmap](#) C7 will include theme support for all controls. I'm sure my beta copy will be here any minute now.

Last but not least, there might be a way to get the [PowerOffice Power-XPTheme](#) third party add-on to work with hand-coded apps, but I haven't pursued that. I do use it with my appgen-based apps.

[Download the source](#)

---

[Mark Riffey](#) has been in the software business in one form or another since the early 1980s. His

background includes software development for two internationally known enterprise software vendors, the world's premier information systems services company and a Fortune 100 manufacturer. After leaving the ranks of the gainfully employed, he bought a nearly-dead vertical-market software company and grew it to the award-winning market leader in its niche. After selling that company in 2005, he now runs [Rescue Marketing, Inc.](#), a marketing, business and technology advisory firm. His business philosophy is simple: Be fair to your customers and yourself; surround yourself with brilliant people who can take a joke, work hard, be a good listener and have a little fun. Mark and his wife Jacki have two boys, Alex and Jonathan. Mark's other interests include Boy/Cub Scouting, backpacking/hiking and almost anything else outdoors, classic blues guitar, golf and photography.

## Reader Comments

### [Add a comment](#)

- [» I tried out your .manifest file and it worked like a charm....](#)
- [» Stan, Yep, that's all it takes. The steps to add it to...](#)
- [» I found that if you simply create an export file \(.exp\)...](#)
- [» Very cool. Thanks for the extra info on...](#)

# Clarion Magazine

## Metadata Management: The Way to a Well-Ordered Enterprise

by Michael Gorman

Published 2006-12-29

No one would argue that enterprises should have their financial books in order. How else can you know where you've been, where you are, and where you project you'd like to go? But just what is the "data" that's in the financial books? It's not real data; rather it's abstract representations of the real data. If it were real data, then you'd actually see inventory data, reserves data, manufacturing cost data, and HR benefits data. But you don't. What you see are abstracted financial representations of these data. That is, when you look at financial statements you're looking at data about these data, or to use the formal term, you're looking at *metadata*.

What then is the financial system that manages all the financial data? It's a *financial metadata management system*. Of course you'll call these systems names like General Ledger. A key measure of the quality of the enterprise is the quality of its financial metadata within its financial metadata management system. You cannot have enterprise-wide, integrated and non-redundant quality financial data about your products, sales, employees and customers without an integrated, federated and non-redundant financial [metadata] management system. They go hand in glove..

Analogously, IT has its set of books and in those books are metadata. As with financial-metadata, you need a metadata management system to manage IT metadata. It also follows that a key measure of the quality of the enterprise is the quality of its IT metadata management system. You cannot have enterprise-wide, integrated and non-redundant quality IT metadata data without an integrated, federated and non-redundant metadata management system. They too go hand in glove.

It's all about quality, and this paper is focused on quality IT Metadata. How important is good data quality? A Google search on the phrase "data quality" turns up more than 2,600,000 hits, largely dealing with data quality problems and solutions. Data quality problems are rooted in discordant semantics, which are, the rules for meaning and usage. For example, what happens if your company lacks policies for:

- Having or not having dashes in social security numbers
- Using 0 and 1 for Gender, vs 1 and 2 for Gender. (Value domain mismatch)
- Name consistency: Mike Gorman vs. Michael M. Gorman (Same person, different names errors)
- Name differentiation: Michael M. Gorman vs. Michael M. Gorman (Different persons, same name)
- Standard formulas: For instance, March East Region Sales = Sales for March of NE Division + Sales of March of SE Division, but, the March Sales of NE Division is Net After Expenses, while the March Sales of SE Division is Total Monthly Sales.

The challenge is not simply whether there are data quality issues, nor how to fix them, but how to design these data quality issues out of the IT process from the very beginning. Not only will enforcing data quality make fixing errors faster, it also will make IT system development faster and cheaper to evolve and maintain.

The way to achieve data quality is to install an infrastructure of quality IT metadata, and with it a quality metadata management system. But before you can do that you'll probably have to convince your boss of the benefits of metadata management.

## Convincing the boss

Any enterprise-wide metadata management system and infrastructure has to have a beneficial impact on the bottom line. Here are four key measures that will determine your success:

1. Improves Quality
2. Improves Productivity
3. Decreases Cost
4. Decreases Risk

If your installed metadata management system cannot deliver these four key measures then you should and will be judged a failure. So, what are you going to do? Here are some

suggestions for measuring the quality of IT metadata.

## Design away ETL

Determine how many programs are being written to essentially perform an extract, transform, and load (ETL) function. For example, if your sales, customer management, ordering, inventory, and billing systems are all "stove pipes" then ETL software is required to bring all that data together to have total customer or product management. If you had data with high quality integrated and non-redundant semantics across these systems and/or if these systems operated off a single integrated database, then these programs would largely be unneeded. Instead of massive quantities of ETL programs, the increased levels of integration can range from integrated semantics across the set of multiple databases and information systems, to an integrated database that all the separately created information systems can access, to a completely integrated system. Yes, programs would be needed for building summaries and the like, but in most enterprises the on-line analytical programs are largely accomplishing that role. The existence of these ETL systems therefore is a direct consequence of not having a quality metadata infrastructure. Compute the life cycle cost of these systems and add this to the Cost of Bad Metadata Management.

## Consolidate and integrate

Determine how long it takes to design a program to process against a database. Does the program have to "fight" with the database's design and build all sorts of temp files, extracts, and the like? All of that is generally due to bad database design. If your enterprise policy is well engineered, that is, integrated, not conflicting, and non-redundant, then so too will be your database designs. If a database is very well designed, then building the essential logic of the program is a code generation step. For all of us, this is a "no brainer" with Clarion. If you are building many database integration programs then you need to consider redesigning your databases to be fewer in quantity and more integrated. In short, consolidate and integrate.

If you've already done that and your programs still "fight with the database" then likely the database's design is wrong. That's either a design problem that needs IT attention, or an enterprise-policy problem that needs corporate-management attention. If your data capture and updating programs are all OK but your reporting programs are a processing nightmare then consider building "data warehouse" databases that contain data from many

other databases and are tuned in design, for example, for just reporting the total customer experience with your business (that is, ordering, delivery, maintenance, returns, and feedback). Figure out the quantity of [function points](#) of these programs and divide each by the quantity of database tables that are being accessed. You should have about 80 function points per table. A higher number means more embedded processes as that's the main source of the counts. A lower number means fewer to essentially no embedded processes. I estimate the average cost to build the function point, if you are using a code generator, to be about \$50.

The very existence of this fight and all the infrastructure that must be created to compensate is a direct result of not having a quality, enterprise-wide, and federated metadata management infrastructure. Compute these costs and take the excess and add it to the Cost of Bad Metadata Management.

## Standardize reference data

Do you have standardized reference data? Reference data is metadata. It is created, managed, distributed, and employed across all databases and information systems in the enterprise either physically or virtually. In the examples above, are there standard "city-state-zip" code tables that are available to all databases and systems? Is there one place for all key customer data such as addresses, contacts, assessments, rankings, and phone numbers? Compute the cost of defining each effectively duplicated set of reference data, and then add all the costs in excess of creating and maintaining reference data more than once to the Cost of Bad Metadata Management.

## Master data

Do you have authoritative data sources? These are now called Master Data. Some organizations also call these data Strategic Data. For example is there one definitive place to which all key customer data updates are driven and then, because of that there's one definitive place from which all customer data can be referenced and/or employed? Master data is just another version of the key slogan "Define once, use many times." Master data's slogan is "create once, store once, and update once to then use many times." Regardless, the process and infrastructure of defining, knowing about, and managing all this master data is metadata management. Thus, if there is more than once definitive source for all multiply-used data, then compute the cost all this data creation and maintenance (beyond the first instance) and add that to the Cost of Bad Metadata Management.

## Assign enterprise identifiers

Do you have enterprise identifiers for all assets that reside in the IT enterprise? An enterprise identifier is a unique number that is not information-bearing, and that is assigned to each asset as it first comes into existence within the enterprise. Is there a central database and supporting metadata management system for maintaining all metadata about these enterprise identified assets? In the above example there would be "master" customer identifiers, product identifiers, employee-identifiers and the like. Once these identifiers are defined and deployed complete knowledge about all uses of customer data is available because the customer is definitively known through the enterprise identifier. Enterprise identifiers are key to the successful creation and deployment of authoritative data sources. If enterprise identifiers are not created and maintained then compute the cost of all the cross reference identifiers, and compute the cost of all the human and computer resources necessary to determine the unique set of assets. Subtract from these costs the costs to create and use a single set of enterprise identifiers. Add the difference to the Cost of Bad Metadata Management.

## Remove redundancies

When the managers get together to build budgets for your business do they argue about what the "numbers" mean? That is, do they have different numbers for the count of employees, total sales, cost of inventory, value of assets, and all that? If they have conflicting numbers, then once the cost of creating all those conflicting numbers is accounted for, compute the cost of all those arguments and the cost of coming to a determination of the correct "numbers." The infrastructure that contains all the definitions, the processes that compute the numbers, their supporting systems, the calendar and business cycles that are operating to compute the results, and of course the interrelationships among all these definitions, numbers, calendars, and business cycles are all metadata and the system to manage it is the metadata management system. Again, compute the cost of all these redundancies, fights, time, effort and energy to resolve the discrepancies, and add that cost to the Cost of Bad Metadata Management.

Now, if you are armed with these numbers then you can quickly say something like this to the CIO:

*If we have quality metadata and a quality metadata management system then we can*

*reduce the cost of all software design and implementation by at least 40%. We can reduce the quantity of data storage by 50%.*

The CIO will listen to your story because you're talking the CIO's language: improved productivity, reduced cost, lowered risk, and increased quality. Metadata management is the key foundation block to a well ordered enterprise just as is its set of well-ordered financial books. It's that simple.

---

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

## Reader Comments

[Add a comment](#)



# Clarion Magazine

## Printing a "No Records" Report

Published 2006-12-22

The request seemed simple. Take a program that is currently generating reports, and modify it so that the reports are generated even if there is no data.

The standard behavior for a Clarion, ABC, report, if there is no data, is to pop up a message which says "No records". Suppressing this message is easy, but the client wanted something slightly different. What he wanted was that the report would still print, but with a data band explaining why there wasn't any data to print.

As with many things ABC, the solution is simple. However it takes a bit of digging to get everything just right. In short, you'll need to create a special report detail band, set it to only display if there are no records, and then manually force the report to print.

### Step 1: Add the detail band

The first thing to do is to create a new band on the report. Call this band `NoDataDetail` (see Figure 1).

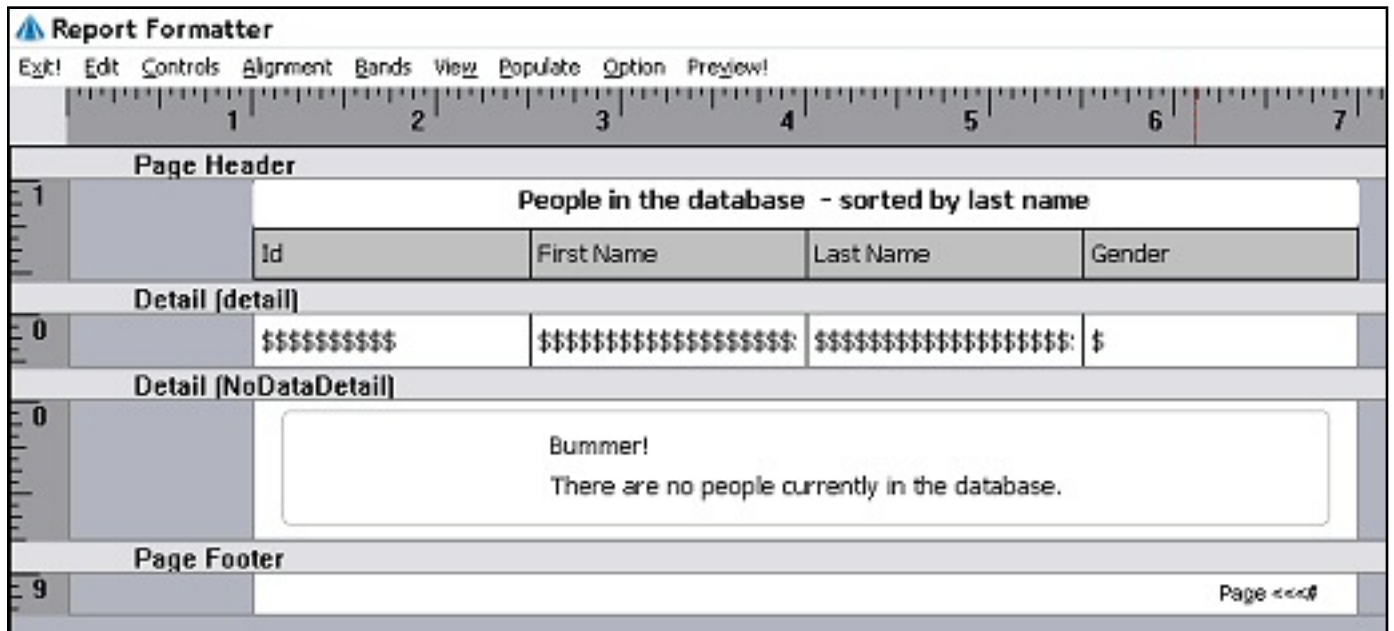


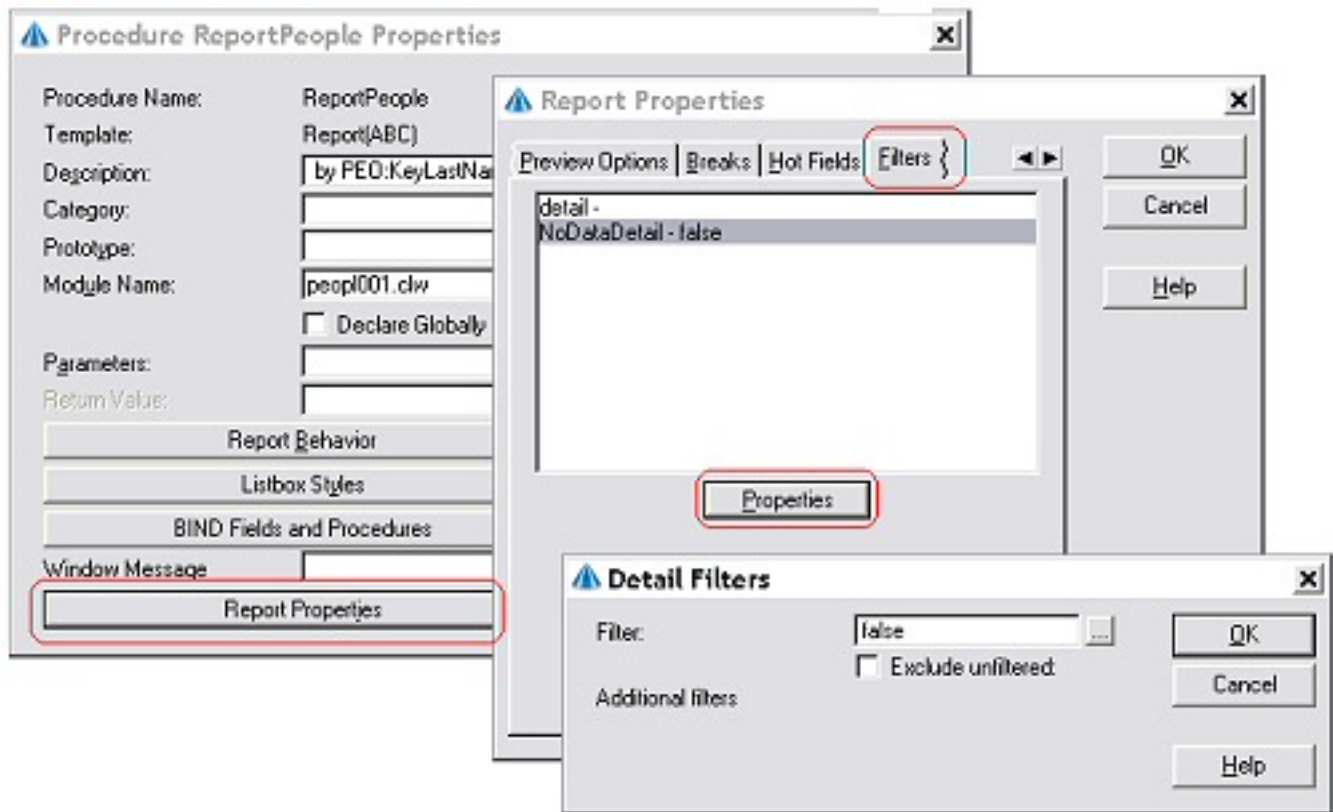
Figure 1. Creating NoDataDetail ([view full size image](#))

## Step 2: Set the band filter

Set the filter for this band to

False

This will ensure that the band is not printed when records are found.



**Figure 2. Setting NoDataDetail's filter ([view full size image](#))**

At first glance the rest should be an easy problem. You can set a filter for each band, so all it then needs is a suitable filter so the band is printed if no others are printed. Unfortunately it isn't this simple. The usual ABC behavior is to check for records first, and if there's no data, then don't open the report at all.

### Step 3: Open the report

The `TakeNoRecords` method is the starting point to the solution. This is the method that is called when no data is present for the report.

In this case though more code is required. The report has to be opened, and the `ReportManager` class (which is the class driving the report) has to be correctly informed that the report is open.

In the `ThisWindow.TakeNoRecords` method, before the parent call, add this code:

#### Suppressing the message

Although suppressing the "No records" message wasn't the requirement in this case, it's not an uncommon request. So if you do want to suppress the message, just put the following in the `ThisWindow.TakeNoRecords` method embed point:

```
Return
```

```
Make sure you put this code in before  
the parent call (i.e. before  
PARENT.TakeNoRecords)
```

```

IF Not SELF.Report &= NULL
  SELF.OpenFailed = 0
  OPEN(SELF.Report)
  IF Not SELF.Attribute &= NULL
    SELF.Attribute.Init(SELF.Report)
  END
  PRINT(Rpt:NoDataDetail)
  BandPrinted = True ! this line is optional, see step 7 below
  IF Not SELF.Preview &= NULL
    SELF.Report{PROP:Preview} = SELF.PreviewQueue.FileName
  END
END
END
RETURN

```

The item in bold is the name of the "no data" detail that was created earlier in step 1.

#### Step 4: Modify AbReport.Inc

Unfortunately, at the time of writing this (Clarion 6.3, build 9055), there is a slight complication. In the line above, `Self.OpenFailed = 0`, the `OpenFailed` property is set to zero. This property is set to one by the class, just before the call to `TakeNoRecords`. The complication is that the `OpenFailed` property is a `PRIVATE` property of the class.

In order to use this property in the app it needs to be set either as `PROTECTED` or as `public`. To do this you need to edit the `\Clarion6\Libsrc\AbReport.Inc` file.

Search the file for

```
OpenFailed BYTE,PRIVATE
```

and replace it with

```
OpenFailed BYTE !,PRIVATE
```

Changing the ABC classes is never a great idea. Ideally you would not need to make the change mentioned in Step 4. In future versions of Clarion it may not be necessary to do so. I strongly suggest that you avoid making changes to ABC whenever possible. And keep in mind that if you do make changes you'll you'll have to redo this modification after

upgrading to a new release.

That said, the nature of the change is relatively safe. It cannot introduce any side-effects to existing programs, and won't affect all your other reports in this, and other, apps. Since the change means making the property *more* visible (not less so) by definition it cannot affect any *existing* code.

The nature of the change is also "visible". If the change is not made, the compiler complains. This is better than a feature simply not working anymore. This means if Clarion 6.3, build 9056 still has this property set as Private, then the compiler will complain, basically prompting you to make the change again.

The risk in modifying the shipping classes is that SoftVelocity may change how the class works and remove `OpenFailed` entirely; after all, you make code private so you have the freedom to rewrite it without affecting how others use the class. But the odds of `OpenFailed` going away are fairly low.

## Step 5: Set `OpenReport`'s result to 0

If there are records for the report, then the `OpenReport` method, in the class, returns a non-zero value. Since code may be added to the `OpenReport` method by other report add-ons, it's important to reset this result to zero. Since the report should always be opened, it's simple to override the `ReturnValue` which is returned by `OpenReport`.

So in the `ThisWindow.OpenReport` method, straight after the call to `Parent.OpenReport` add the following line;

```
ReturnValue = 0
```

## Step 6: Set `SELF.Response`

The only thing left to do is to set the property `Response` to `RequestCompleted`. This is best done in the `AskPreview` method, before the parent call.

```
SELF.Response = RequestCompleted  
PARENT.AskPreview
```

The report will now print if no records are found.

## Other considerations

It's worth making sure that the report accurately detects that there are no records to print. In other words, `TakeNoRecords` only gets called if there is no data for the report. It doesn't get called if there is data for the report, but no bands are printed.

My point here is that you can add filters to all your bands, and even though there is data being processed, because of the band filters nothing is printed. In this case you'll get a blank report (with just the header and footer) but no `NoDataDetail` band.

One solution to this is to make sure that records are properly filtered using the template options and, if necessary, using the `ThisWindow.ValidateRecord` embed point. If you return from `ValidateRecord` (setting `ReturnValue` to either `Record:Filtered` or `Record:OutOfRange`) then `TakeNoRecords` will still be called.

## Step 7

If filtering the data completely in `ValidateRecord` is not possible, and you must use band filters, then you need to add some code to handle the case of a blank report. In this situation there is data, but because of the band filters, no bands are printed. This is more likely with reports with complicated band filters.

If you think you are going to encounter this situation, then the solution is a two step process. First you need to set some local variable to true the moment any band is printed. Second, in the `EndReport` method, before the parent call, test this variable. If it is false, print the `NoDataBand`.

For example, assuming the variable is called `BandPrinted`, declared as a `LONG`, then the code in `ThisWindow.EndReport` would look like this;

```
If Not BandPrinted
    Print(RPT:NoDataDetail)
End
```

## Summary

Suppressing a "No records" message in a report is easy; printing a "No records" report is a bit more complicated, but still relatively straightforward provided you don't mind making (and maintaining) a simple modification to one of the standard ABC classes. And you may need to add a bit more code if your detail band filtering removes any records that would otherwise have printed.

[Download the source](#)

## Reader Comments

[Add a comment](#)

# Clarion Magazine

## A Callback Technique for Capesoft's FileManager2

by Randy Rogers

Published 2006-12-21

CapeSoft's [FileManager2](#), and its more recent incarnation [FileManager3](#), are amazing products: they completely automate the process of upgrading client databases, whether local or across a network, and make it easy to do file maintenance. I can't imagine anyone today not using FileManager with TopSpeed file based applications; particularly on projects with the potential for many database changes. But because FileManager does so much, startup time on large applications can get lengthy. In this article I'll show you how to use a callback technique to provide user feedback during FileManager2 startup.

My company, [Keystone Computer Resources](#), was started in 1991 and developed a small vertical market application for local municipal governments. The product was TopSpeed file based and, as with any new application, went through many database modifications. These database modifications and conversions were a real pain to deal with until I discovered and implemented Capesoft's FileManager2 product. Over the last fifteen years the municipal software has evolved into a suite of over 16 executables and 30 libraries and the database has grown from 30 to 220 files.

As the size of the application grew my customers began to notice degradation in performance at start up. I did everything I could think of to solve the slow start up problem. All the proper opportunistic locking settings had been made on the servers and workstations. The global application file settings were correct and the application suite and libraries made use of a single global data DLL which contained the FileManager code. I tried to explain to my customers that the new runtime library was doing more work at startup to ensure database integrity. I even created a small application that would display a progress bar as the libraries got loaded.

Although this was not perfect, it did ease the wait by keeping the user visually stimulated with feedback. All of the executables needed to load the global file DLL, the one with FileManager2. Further investigation revealed a noticeable delay in the loading of this global DLL. If I removed FileManager2, the DLL loaded quickly; the delay was occurring during the FileManager2 initialization code. I am not sure what the FileManager2 is doing during this process, but I had to



find a better way to deal with the delay; I could not simply remove the FileManager2 as I rely heavily on its functionality.

My solution was to create a separate, self contained, executable that was built from the main dictionary. The executable (dbupdate) would be run first if there was a version change in the software. The dbupdate program would open and close all the files in the database to force the FileManager2 to do its magic on the files that had changed. I also removed the FileManager2 from my global data DLL. This made a tremendous improvement in my application load times. Taking this approach had the added advantage of allowing me to add conversion code to the dbupdate program to handle any data initialization or conversion requirements for the new version.

This solved the problem for the main applications suite, but, the dbupdate executable still had this annoying delay at initialization. I felt I had to supply the user with feedback during the initialization phase of the dbupdate program, which is why I needed a callback from the FileManager2.

What follows is a discussion of how I implemented the necessary callback to provide feedback to the user during the FileManager2 initialization loop.

## Creating the callback

There were basically two problems to overcome. First, I needed to isolate when the FileManager2 performed its initialization code. In the standard implementation, this occurs before the window is opened. I really needed to have this happen after my progress window was opened. Secondly, I needed to somehow get notification or control after the FileManager2 initialization for each file occurred.

Fortunately, the first problem was easy to solve. I really wanted to have the initialization code in its own procedure; that way I could have it execute when I wanted. The FileManager2 global extension template has an advanced tab and VLDS (Very Large Dictionary Support) that allows the user to break the initialization calls up into procedures ... perfect! I could provide the name of a FM2 init procedure and then use another FileManager2 template to complete the moving of the initialization code to a procedure.

Since I specified `FM2Init` as my procedure name in the FileManager2 global extension template, I needed to create a procedure of that name in my application. I created a source procedure and added the FM2 DLL Init function Code template at priority 4000. The template prompts for a Filename range:

That completed the movement of the FileManager2 Initialization code to a procedure. The call to the procedure will still occur prior to opening my progress window, so I needed to devise a way to control when the procedure got called. I decided on a global flag, `glo:bInitFilemanager` to control the calling of the procedure.

I created a BYTE variable named `glo:bInitFileManager` in the Global Data of the application. The default value is set to false and then gets set to true after I open my progress window.

In the FM2Init procedure I added the following embed code at priority 2000:

```
IF glo:bInitFileManager = TRUE
```

And at priority 7000

```
END
```

Now the code inside the procedure will only execute if the flag is set to TRUE.

In the procedure that has the progress bar, I added the following code after the parent call in the OpenWindow event embed:

```
glo:bInitFileManager = TRUE
FM2Init()
```

The second problem was to get the FileManager2 initialization code to call back to a procedure of my own design after it initializes each file so that I can update my progress bar. I also needed some global variables to communicate between the procedures. In the Global Data I added the following:

```
glo:FilesProcessed          LONG,AUTO
glo:FilesToProcess          LONG,AUTO
glo:feqProgressBar          LONG,AUTO
```

The FileManager2 templates provide two user embed points. This one allows the user to place code before any FileManager initialization code:

```
#EMBED(%FM2Init,'FM2: Start of Initialisation section')
```

and this one allows the user to place code after the FileManager initialization code has completed. (note: the loop referred to is a template #FOR loop, not a Clarion code loop):

```
#EMBED(%FM2EndLoopFM2InitCode,'FM2 - Just outside Init code loop')
```

Neither one of these embeds provided what I needed, an embed before or after the initialization of each file controlled by the FileManager. A more thorough look at the FileManager2 template revealed the following hidden embed:

```
#EMBED(%FM2InsideLoopFM2InitCode,'FM2 - Inside Init code loop'),HIDE
```

This embed allows a template to place code after the FileManager initializes each file.

I created a template named FM2Callback.tpl that would place a call to my procedure in this hidden embed (line breaks added).

```
#EXTENSION(FM2InitCallback,'FM2 Initialization Callback'),PROCEDURE
#BOXED('FM2 Callback')
  #DISPLAY ('This template adds a callback procedure
           to the FM2 Init loop. '),AT(10)
  #DISPLAY (' ')
  #PROMPT ('Callback Procedure Name:',PROCEDURE),
          %CallbackProcedure,DEFAULT('FM2InitCallback')
  #DISPLAY (' ')
#ENDBOXED
#AT(%FM2InsideLoopFM2InitCode)
%CallbackProcedure('%File')
#ENDAT
```

In this example, the callback procedure needed to be prototyped with a string parameter, the name of the file just processed.

Finally, I needed to create my FM2InitCallback (template default name) procedure. This procedure updates the progress bar during the FM2 initialization loop. I created a source procedure and added the following to the processed code embed:

```
glo:FilesProcessed += 1
IF glo:FilesProcessed <= glo:FilesToProcess
  glo:feqProgressBar{PROP:Progress} = INT(100 *
(glo:FilesProcessed/glo:FilesToProcess))
  DISPLAY(glo:feqProgressBar)
END
```

The FM2InitCallback procedure makes use of global variables, initialized by the progress bar window procedure, to set the progress bar width. In order to do this properly, I needed to know how many files would be processed, so I created a code template to get this information for my application. From the FM2Callback.tpl file:

```
#CODE(GetTPSFileCount,'Get Count of All TPS Files in the Dictionary')
#PROMPT('Target Field:',FIELD),%TargetField,REQ
#DECLARE(%FileCount)
#SET(%FileCount,0)
#FOR(%File)
  #IF(%FileDriver = 'TOPSPEED')
    #SET(%FileCount,%FileCount+1)
  #ENDIF
#ENDFOR
```

```
#INDENT(-1)
%TargetField = %FileCount
#INDENT(+1)
```

I placed this code template prior to opening my progress window and specified `glo:FilesToProcess` as the target field. Finally, I initialize the `glo:FilesProcessed` variable to 0 and set the `glo:feqProgressBar` variable to the `feq` for my progress control. Now, as the `FileManager2` performs its initialization, a nice progress bar is displayed for the user.

You will probably only notice the delay when opening a large number of files across a network, and, in such cases, it is nice to keep the user informed of the initialization progress. But even if you don't have a noticeable delay some of the techniques presented may be of value. I have created a small application and provided the templates to accompany this article to demonstrate the problem and the solution.

To see all this in action make sure you have `FileManager2` installed, then do the following:

1. Register the `FM2CallBack.tpl`
2. Compile the `LargeDct.prj` and run the resulting executable from a folder on the network. This will create 200 Topspeed files in the network folder.
3. Compile and run the `Example.app`. When prompted for the network folder, select the folder containing the 200 TPS files created earlier.

### [Download the source](#)

#### CapeSoft comments and Randy's responses

*Nothing in life comes for free, and as Randy has noticed there is a delay on startup as the necessary FM2 initialization code does its magic. Randy's solution (of creating a separate dbUpdate program) is often suggested. Randy's implementation here is especially good in that he goes a step further even in the dbUpdate program. However it does suffer from some drawbacks:*

*1. If the file structure does change, and this dbUpdate program is not called, then the user will get an error 47. This would most likely happen if data from a backup, or another site, were introduced into the data set. At this point the dbUpdate program would need to run, and all should be fine.*

**Randy:** In our case, this is all controlled by our updating process

*2. The dbUpdate EXE has to remain in sync with the actual program Exe. Again this is probably not a problem for the programmer, as proper care will ensure that your install is always correct. However if the user (or upgrader) backs up just the one exe and the two get out of sync then there can be problems.*

**Randy:** Again, this is not an issue in our application as we control the backup/restore processes. If the users do something outside our application, then it is their issue and our response is to restore the last full backup.

*3. He's talking specifically in the context of FM2 here, i.e. TPS files. I wouldn't recommend this approach to SQL*

*users without testing first.*

**Randy:** Exactly, this is for a database containing a large number of TPS files located on a network server.

*4. His solution is limited to programs that have a single data set (ie where each table in the dict matches 1, and only 1, TPS file on the disk). If backup data files are restored into the data folder, then dbUpdate needs to be re-run.*

*5. His solution reduces, or removes, the other functionality in FM2. Specifically the Runtime File Manager, and AutoFix would have complications.*

**Randy:** I have never used RFM or AutoFix .

*Here are some suggestions that might take this approach one step further. Please note that all these suggestions have potential pitfalls in specific circumstances. In many cases the possibility of a problem goes up, and if things get out of sync [ multiple data sets / backup's restored ] then problems can occur.*

*1.) File Manager 2 has a setting on the Advanced tab called "Open UPG Once". This reduces the overhead during the startup, especially over slow networks. [ If you send me Randy's example I'll run it with this switch on and off to compare results. ] [ This is a fairly safe option, the risk of problems here are very small. ]*

*2.) Add the FM2 extension back in the data DLL, but use the VLDS support mentioned above. In this procedure store the date, and time, of the current exe in an ini file. On startup compare the date & time to see if the Fm2 initialization is necessary. If not, skip over it. If this code fails for some reason (for example if a backed-up data set is restored) then an error 47 is likely. [Thus this is not recommended for sites that have multiple-data sets, and it can lead to problems if a data set is restored without the ini setting being reset.]*

*Randy also includes code to force each file open, and closed so that if a conversion is necessary, then it occurs here. This is also something that could be safely avoided if the exe has already run.*

**Randy:** This only happens when dbupdate is run, and it is run only when needed.

*3.) An extension of suggestion (2) above. Fundamentally when a new exe runs for the first time it adds any new file structures to the UPG file. It's very important this happens because without it conversions are impossible. It also checks that the current version in the UPG file matches the current version in the exe. However as long as the EXE does not change, it should not be necessary to re-do these checks. If you are pretty sure the check is not necessary then there is a command line switch /LOADFAST which can speed it up. Tip: use SETCOMMAND to programmatically set a command line switch.*

**Randy:** Our mechanism handles new files properly; I was not aware of the /LOADFAST switch

*4.) Even more out there: what happens if the VLDS procedure is run on a separate thread during startup? I've never tried this and it would be an interesting approach. It would only work though if it completed before the program tried to open a single file.*

**Randy:** Well, that is kind of what our dbUpdate does.

The example accompanying the article, as provided, does not suffer from any of the potential side-effects described by Bruce. But as always the reader should test to his/her own satisfaction.

[Randy Rogers](#) is a data processing professional with over 35 years of experience in a wide variety of industries including accounting, municipal government, insurance, printing, and pharmacoconomics. He has a degree in Mathematics from Florida State University and is the president of [Keystone Computer Resources](#). Randy is the author of [ClassViewer](#), a utility for browsing the Clarion class hierarchies. He is also the creator of NetTools, Queue Edit-in-Place, and Screen Capture Tools for Clarion application developers.

## Reader Comments

[Add a comment](#)

- [» Great ideas! We have been doing a similar update process...](#)

# Clarion Magazine

## No! You Can't Cancel

by Steven Parker

Published 2006-12-13

Clarion developers are a nasty bunch. They do not want to let their end users cancel forms. At least it seems so from the frequency with which questions are asked about trapping the Escape key or Windows' red X.

In many cases, the desire to stop a user from exiting a data entry form without saving is due to ill-considered design. For example, I have a form on Purchase Order header records that has a tab on it, and on that tab a user may add, change or delete records in the Purchase Order detail file (I plead "the boss made me do it").

What's the big deal? Glad you asked. Suppose a user starts a new Purchase Order (creating a parent record) and adds items to the PO (creating child records). If there were a way for the user to escape from the form, the child records would have been saved but the parent record would not be. Can you say "hosed files?"

In this case, I should have done all my work in memory. If the user accepts the form (presses the Ok button), write the queue or In-Memory Database Driver records to file. If the user cancels, discard the queue or IMDD file. To handle inserts or deletes of existing parent-child (header-detail) combinations, there is quite a bit of detail to attend to. In any case, I could:

- dot my i's and cross my t's
- just use Mike Hanson's [SuperInvoice](#)

or,

- try to prevent the user canceling when any change has been made to the child file.

Management chose the last alternative. Truly, I am innocent!

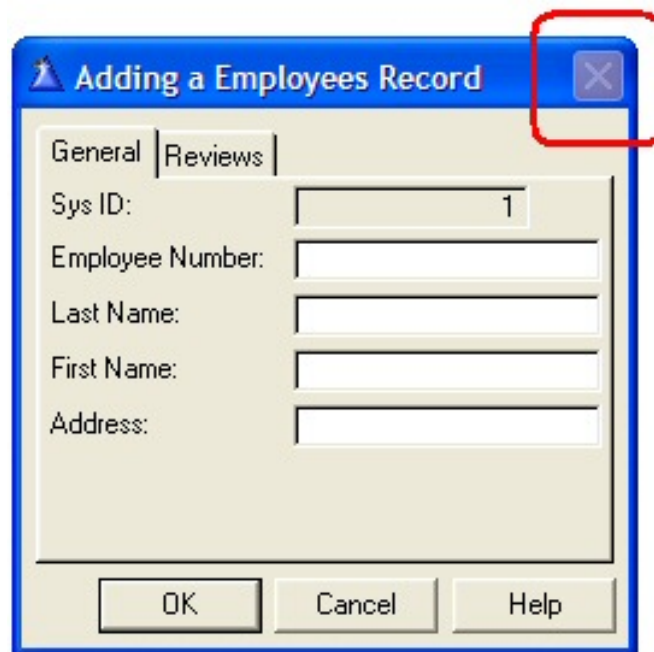
There are three ways that I can think of to stop a user from canceling a form. Of course, I presuppose

that there is a variable or condition that is set and tested to determine whether a cancel is allowable before implementing any of them. In the sample app, downloadable at the end of this article, I do not set up any testable condition; be warned.

## Disable the X

Microsoft, once again, in its infinite wisdom provides a red X in the upper right corner of each window. Clicking on this closes the window and by-passes all the validation built into the templates. Clarion template-generated procedures do not trap this.

The solution to disabling the X comes from (who else?) Jim Kane. In the sample app, from the Frame, click Browse | Browse Employees file – no X. Then press Insert and you will see the form in Figure 1.



**Figure 1. Red X disabled on Form**

This is how Mr. Kane did it:

In the global data area, add the following equates:

```
SC_CLOSE           equate(0f060h)
MF_GRAYED         equate(1)
```

In the global map, prototype the following APIs:

```
Module('win32')
    EnableMenuItem(signed hMenu, signed uIDEnableItem, signed uEnable),|
    bool,pascal,PROC
```



```
DrawMenuBar(signed hWnd),bool,pascal,PROC  
GetSystemMenu(signed hWnd, bool act),signed,pascal  
End
```

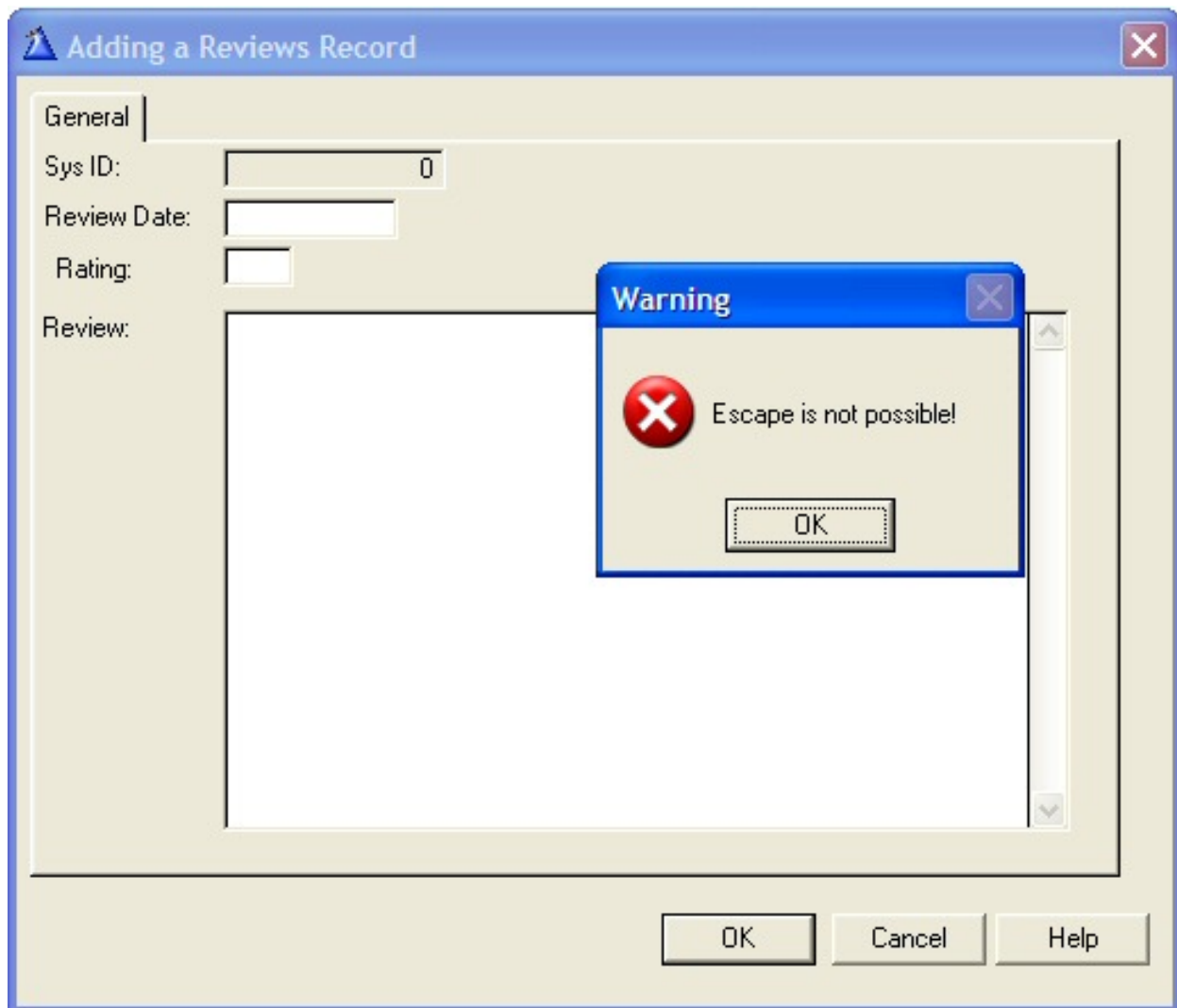
In the form procedure, declare hMenu as a local long. Then after the window is opened:

```
hMenu = GetSystemMenu(ThisWindow{PROP:Handle}, False)  
EnableMenuItem(hMenu, sc_close, MF_GRAYED)  
DrawMenuBar(ThisWindow{PROP:Handle})
```

As Mr. Kane notes, this will disable the Windows close option but the Esc key and ALT-F4 will still close the window.

## Trapping ESC

In the sample app, from the Frame, click Browse | Browse the Reviews file (Wizard) – no ESC Then press Insert. Press the escape key:



## Figure 2. Trapping the ESC key

Here is how this is done.

On the window, right click and select Alert... and click Add. When the Input Key window comes up, press the ellipsis and type `EscKey` in the Enter Key prompt.

Then, in Window Event handling for the Alert key:

```
If KeyCode() = EscKey
    Message('Escape is not possible!', 'Warning', Icon:Hand)
    Select(1)
    Cycle
    Return Level:Notify
End
```

will prevent the user from closing the window on the Esc key.

But the Cancel button still works.... So, not even combining trapping the X and trapping the Esc key will fully cover the need.

Besides, using both techniques begins to involve a lot of code.

## TakeCloseEvent

Any time a window is closed, its `TakeCloseEvent` method is called. Therefore, this is an *ideal* place to test a "do not close" condition. For example:

```
If MyCondition = True
    BEEP(BEEP:SystemExclamation) ; YIELD()
    MESSAGE('Cancellation is futile.', |
        'Cancel Not Permitted', ICON:Exclamation)
    Return Level:Notify
End
```

In the sample app, Browse | Browse Reviews (Adjusted) – Late Trap. Before going into this update form, I suggest you make sure you have Windows' TaskManager ready. I initialized `MyCondition` to `True`, so you will never get out of the form without killing it from TaskManager.

Change the code (the app is C5.5) and change this embed so that it reads:

```
If MyCondition = True and Self.Response = RequestCancelled
```

and you may have better luck closing the form window. With this modification, only canceling is

barred and this is what is wanted: the Esc key, the Cancel button and the red X will not close the form.

The success of this approach depends on correctly setting the condition(s) to check....

## Summary

I knew I should have designed my purchase order parent-child update so that it was done using queues or, later, the In-Memory Database Driver. Inserts would have been easy; if the user pressed Ok, dump the memory buffers to file. Changes would have been a bit more difficult since I would have to re-write the header record but remove all child records before dumping the new ones to file.

Of course, there is also the issue of autonumbered keys. In my case, the Purchase Order header file has one.

But, this is what management wants. Yes, yes, I should have fought it.

Yet, given that I didn't, the templates still forgive me. I have several ways to stop users from hurting my files.

[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 version 2007 (DOS, that is). He reports that, so far, Clarion is winning. Steve has been writing about Clarion since 1993.



## Reader Comments

[Add a comment](#)

- [» With this modification, only canceling is barred and this...](#)
- [» Okay, perhaps it might have been better stated as something...](#)

# Clarion Magazine

## A Customized Deep Assign Function

by Alan Telford

Published 2006-12-12

Programmers are fascinating creatures. When their curiosity is aroused they will divert from whatever they're doing to answer that call. I was struggling with an SQL statement recently so I asked a colleague for help. She couldn't give me an answer right away so I went back to struggling. But ... I had aroused her curiosity, and 15 minutes later the answer was in my email box and even included an apology for taking so long.

Later that week it was my turn to be challenged. Another colleague had a procedure `SomeProc procedure(file pSource, file pTarget)` and needed to copy all the fields which were the same from file `pSource` to `pTarget`. "Easy", I told him and sent him on his way with this code snippet.

```
! declare a group reference to record buffer
sourceRecord    &Group
targetRecord    &Group

! assign the reference
sourceRecord &= pSource{prop:record}
targetRecord &= pTarget{prop:record}

! now use deep assign to copy the similar fields
TargetRecord ::= sourceRecord
```

Several minutes later he returned. "It didn't work." The deep assign from a group reference to another group reference failed. As he only had a small number of files he had to support, I gave him a temporary (but unsatisfying) workaround:

```

if pSource &= File1 and pTarget &= File2
  File2.record :=: File1.record
elsif pSource &= File2 and pTarget &= File1
  File1.record :=: File2.record
! add other combinations here
End

```

But I could not go back to work. The power of a good challenge was doing its work.

## Rolling my own deep assign

I was already familiar with WHAT and WHO from writing my [ExportQtoCSV](#) function, so I figured all I had to do was loop through both files reading their record structures, find which fields had the same names (using WHO), retrieve the field reference into an ANY (using WHAT), save the matching pairs to a queue, and finally use my matching pairs list to actually copy the *like* fields.

### Step 1: Design the data

I like to start with designing my data structures. I decided for simplicity that I didn't need to save every table field in my queue, but only the matching fields. This led me to the following:

```

matchfieldQ      queue
sourceNdx         long
sourcename       cstring(50)
sourceAny        any
targetNdx        long
targetname       cstring(50)
targetAny        any
end

```

The queue holds one record for each pair of matching fields.

- `sourceNdx` is the position of the matching field in the source record.
- `sourceName` is the name/label of the matching field.
- `sourceAny` is the reference to the field.

The Target file fields are named similarly.

## Step 2: Find the Matching Pairs

Now that I have the data structure it's time to write the code to find the matching pairs. This code will only be executed once before the main file loop, so it should focus more on readability than speed.

The Clarion help section on "FILE Structure Properties" shows me two useful properties: `File{prop:record}` to get a group reference to the file's record buffer, and `File{prop:fields}` to return the number of fields declared in the RECORD structure.

```
! get the reference to the record structure
sourceRecord &= pSource{prop:record}
TargetRecord &= pTarget{prop:record}
! get the number of fields in the record
sourceFieldCnt = pSource{prop:fields}
targetFieldCnt = pTarget{prop:fields}
```

Then to match the fields I have a loop within a loop. The outer loop retrieves the name of each field in the Target file while the inner loop attempts to find a matching name from the Source file.

```
loop targetndx = 1 to targetFieldCnt
  targetname = WHO(targetRecord, targetndx)
  loop sourcendx = 1 to sourceFieldCnt
    sourcename = WHO(sourceRecord, sourcendx)
    IF sourcename = targetname
      !save matching pair in matchfieldQ
      clear(matchfieldQ)
      matchfieldQ.sourceany &= null
      matchfieldQ.targetany &= null
      matchfieldQ.sourceNdx = sourcendx
      matchfieldQ.sourcename = sourcename
      matchfieldQ.sourceAny &= what(sourceRecord, sourcendx)
      matchfieldQ.targetNdx = targetndx
      matchfieldQ.targetname = targetname
      matchfieldQ.targetAny &= what(targetRecord, targetndx)
      add(matchfieldQ)
      break
```

```

        end
    end !loop sourcendx
end !loop targetndx

```

This is only a two-level loop, but you may notice one of my coding habits on the last two lines. I like to identify the end statement so I know what statement block it belongs with. I'm sure looking forward to the new Clarion IDE with support for code folding!

Surprise surprise, the code next exports the queue to Excel so I can see what it contains (refer to my [Debugging Queues with Excel](#) article).

```
ExportQtoCsv(matchFieldQ, 'matchfieldQ.csv')
```

I compiled and ran my test program and expectantly looked for the CSV file but it was not there.

### Step 3: Debug the program

Reluctantly I changed to debug mode to track down the problem. I quickly discovered that `WHO()` returns `pre:name` where `pre` is the file prefix. Each file has a different prefix and therefore none of the field names were matching. I added a `StripPrefix()` function to remove the prefix from the field name. Having solved the problem of different prefixes I also used the `UPPER()` statement to prevent the related problem of case sensitivity.

```

stripPrefix      procedure(string pName) !,string
ret  cstring(50),auto
pos  long,auto
code
pos = instring(':',pName,1,1)
if pos
    ret = upper(clip(sub(pName,pos+1,len(pName)-pos)))
else
    ret = upper(clip(pName))
end
return(ret)

```

I modified the `targetname` and `sourcename` assignments as follows:

```
targetname = striprefix( WHO(targetRecord, targetndx) )
sourcename = striprefix( WHO(sourceRecord, sourcendx) )
```

With much satisfaction I noticed that the CSV was now produced, and I could verify that my `matchFieldQ` had the correct list of matching field names.

#### Step 4: Finalize and test

Now that I had a list of matching fields it was straightforward to assign the matching fields. I've included a basic file loop without error handling for illustration.

```
set(pSource)
loop
  next(pSource)
  if errorcode() then break.
  loop ndx = 1 to records(matchfieldQ)
    get(matchfieldQ,ndx)
    matchfieldQ.targetany = matchfieldQ.sourceany
  end
  add(pTarget)
end !loop pSource
```

At this stage a bit of *test design* is a good idea. Some quick thinking gives the following set of tests to run:

1. Copy source to an identical target file.
2. Change the order of fields in target file.
3. Change the order of fields in target file, and delete some fields.
4. Change the order of fields in target; delete some fields; add some unique fields.
5. Add an array field into source and target record.
6. Add a group structure into source and target record.
7. Repeat tests for different field types (String, Byte, Long, Decimal etc.)

I expected test 1-4 to all pass which they did. But I was rather surprised that tests 5-6 also passed. I'm unsure whether they would work for different size arrays, or differing record structure so I would suggest more exhaustive testing if these elements are important to you. Although I do use arrays in queues, I try not to use them in files, so tests 1-4 were the important ones for me.



## Converting to a class and include file.

Now that my curiosity was satisfied and I had likewise conquered the challenge (although it did take me much longer than the 15 minutes my colleague spent on my coding problem) it was time to make this code usable by others. Which approach should I use: A class or a procedure?

The deep assign needs to save the list of matching fields for reuse. It has two distinct methods (a one time discovery of matching fields, and a repeated assigning of like fields). It can probably be extended with more useful methods in the future.

Everything indicates that a class would be more appropriate, and you can find the MTDeepAssignClass source files in the downloadable zip.

To use:

1. Save the INC/CLW files into the Clarion6\Libsrc folder, and refresh the ABC include files (the easiest way is to exit and restart the Clarion IDE).
2. Include the file at the *module* or *global* level (it will GPF if it's included at the local procedure level).

```
INCLUDE( 'MTDeepAssign.inc' )
```

3. Declare a local object of this class (usually within procedure or routine data).

```
MyDeepAssign      MTDeepAssignClass
```

4. Call the `init` method before your main file processing loop:

```
myDeepAssign.init(sourceFile, targetFile)
```

5. Call the `assign` method inside your main file processing loop:

```
myDeepAssign.assign( )
```

Steps 2 and 3 can be easily handled if you have either of the following:

- [ABCFree templates](#) – use the ClassDeclare extension template.
- [Handy Tools HNDTools template](#) – use the EmbedObject extension template.

If you're using Legacy templates, then replace `mtdeepassign.inc` with `mtdeepassign_noabc.inc`. You can also test the class out with the enclosed C6 example application and dictionary.

[Download the source](#)

---

[Alan Telford](#) has been programming in Clarion since 1994. He is the Chief Software Developer at [Maxtel Software Ltd](#), a New Zealand software company specializing in writing back office computer solutions for McDonald's Family Restaurants and other similar markets.

## Reader Comments

[Add a comment](#)

- » [Unfortunately WHAT in C5.5 is broken, after some random...](#)
- » [I didn't know that about C55. I must say that all my apps...](#)
- » [This was exactly what i needed. And your example app works...](#)
- » [I can think of two reasons you may have this error. If...](#)
- » [My failure was missing understanding af 'File' as type. I...](#)

# Clarion Magazine

## How To Run Multiple Instances of Clarion (video)

by **Richard Rose**

Published 2006-12-12

Richard Rose's Camtasia presentation on how to run multiple instances of Clarion, by special arrangement with the UK Clarion User's Group. This is a 14 minute, 63 meg AVI file. The zip file includes registry files and batch file.

Please note that although you can have multiple instances open, you can only have one instance compiling/generating at one time.

If you have trouble downloading please delete any partial downloads and do a Save As.

Please post any comments below.

[Download the video](#)

### Reader Comments

[Add a comment](#)

# Clarion Magazine

## The ClarionMag Blog

Get automatic notification of new items! [RSS feeds](#) are available for:

**XML** [All blog entries](#)

**XML** [All new items, including blogs](#)

### Blog Categories

- » [All Blog Entries](#)
- » [Clarion 7 Clarion.NET](#)
- » [Future Articles](#)
- » [News flashes](#)
- » [Nifty Stuff](#)

## Merry Christmas and Happy Holidays!

[Direct link](#)

Posted Friday, December 22, 2006 by Dave Harms

It's just about time to close down the office for the Christmas break (although I'll be keeping an ear to the ground for any late-breaking news). There are still a couple of

December articles waiting in the wings, including Mike Gorman's explanation of what metadata is, and why you should be managing yours. Look for those next week.

I'll be back in the office on Wednesday (Tuesday is [Boxing Day](#)). Merry Christmas, Happy Hanukkah, and best holiday wishes to all who celebrate the season of peace and light!

Dave Harms

## C7, Clarion.NET in final testing

[Direct link](#)

Posted Thursday, December 07, 2006 by Dave Harms

SoftVelocity's Bob Foreman has posted a [blog entry](#) stating, among other things, that "we are currently in the final stages of testing for the upcoming Clarion 7 and Clarion.NET releases." This is the first official news (at least that I can recall) on the C7/Clarion.NET release schedule since the [roadmap](#) was published in August. Here's the scoop on those products, from the roadmap:

### **2006 – “Clarion Spirit”**

The next release of the Clarion Win32 product family, code named “Spirit” will further advance developer productivity through enhancements in Code Generation, Code Completion, Code Templates, Data Dictionary functionality, and support for Multiple Active Clarion Systems (MACS). Significant User Interface enhancements have been implemented in the Win32 RTL, including full XP Theme support for all controls, Office 2003 styled Menus, new support for Tabbed MDI windows, and improvements aimed at better support for Unicode (double-byte character sets). A phased approach to Unicode support will continue in the subsequent Comanche release. To see a live demonstration of some of the new UI improvements in Spirit you can view the video .

Performance optimization has been a focus item for this release cycle. You'll notice improvements in the Spirit IDE for Code Generation and

compilation times. An all new high-performance non-blocking multithreaded Clarion IDE Applet task manager has been implemented.

Spirit is currently scheduled to be completed and released this year (2006). It will support development of Win32 applications using any version of Clarion from 1.5 to 7.0. We're considering releasing tech preview builds to Core Subscription users as we get closer to RTM, and before that organizing an Alpha release team, primarily to allow 3 rd party vendors to prepare new editions of their products.

## **2006 – “Clarion Hidalgo”**

The release of the Clarion.Net product family, code named “Hidalgo” is based on the .NET 2.0 Framework. Hidalgo will open a new world of productivity for .Net development through Code Generation, Code Completion, Code Templates, and Data Dictionary functionality. Hidalgo makes all of the functions of the current Clarion RTL including built-in Reporting and standard Clarion File access are available with the exception of Window structures and the ACCEPT construct. For development of desktop applications Window structures are replaced with .NET WinForms. Migration to WinForms is eased by new Template support to convert Window structures to WinForms, as well as additional tools to convert Windows that are not contained within the .APP format into WinForms.

The Clarion.Net compiler will support building applications using WinForms, WebForms (ASP.Net) and Mobile Devices using the .NET Compact Framework (CF). The IDE designer support for CF in this release will likely have some limitations.

The current plan is to release Hidalgo in calendar year 2006. Prior to the RTM for Hidalgo we're considering releasing Clarion.Net subscriptions and tech preview builds, but that will depend largely upon how quickly we can get the heap of new language syntax documented, and the new Template sets tested.

Having been burned on predictions in the past I'm not going to say exactly when we'll see these products released, but certainly Bob Foreman's comment appears to be a light at the end of a very long tunnel.

Oh yeah, most of Bob's announcement has to do with a new SQL training course.

