

Clarion Magazine

Clarion News

- » [xReportPreview 3.6](#)
- » [FullRecord 1.80 Beta](#)
- » [BackItUp 1.9.0](#)
- » [Rasp 0.2](#)
- » [Clarion Desktop 2.61](#)
- » [Free Vista Icons, Wallpaper](#)
- » [New CircoCyan Icon Set](#)
- » [BackItUp 1.8.0](#)
- » [Clarion Daily News](#)
- » [Vivid Help Conductor Template Updated](#)
- » [BackItUp Updated](#)
- » [SetupBuilder 4.x Upgrade To SetupBuilder 6.0](#)
- » [Gitano Source Code Deal Ends This Week](#)
- » [xNotes Browse-Form 1.3](#)
- » [NeatMessage 2.10](#)
- » [Advertising on Clarion Desktop!](#)
- » [SetupBuilder 6.0 Build 1750](#)
- » [Next UKCUG Meeting](#)
- » [2007 Aussie DevCon and Capesoft Training](#)
- » [Capesoft MailSync Announced](#)
- » [CapeSoft World Tour 2007](#)
- » [Clarion Desktop 2.50](#)
- » [Orlando Web Hosting](#)
- » [Huenuleufu Alternate URL](#)
- » [CHT January Newsletter](#)

- » [Gitano January Special](#)
- » [Clarion Third Party Profile Exchange Updated](#)
- » [WinEvent 1000](#)
- » [RPM Add-On Invitation](#)
- » [UltraTree Premium, Up & Up Savings](#)
- » [WindowID 1.10](#)
- » [BackItUp Updated](#)
- » [BackItUp 1.5.0](#)
- » [WindowID 1.10 Beta](#)
- » [Roadmap Update Updated](#)
- » [Smart-Type Version 2](#)
- » [BackItUp 1.4.0](#)
- » [Code Commentor 2.0.3](#)
- » [AFE v2905 Server Upgrade Released](#)
- » [Code Commentor 2.0](#)
- » [Abspoel.nl Clarion Site](#)

[\[More news\]](#)

- » [Clarion 101: Designing A Database](#)

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

Latest Free Content

- » [Clarion Magazine Wants Your TXAs!](#)

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

Clarion Sites

Clarion Blogs

“ A top rate e-magazine. ”
▶ Julian Burke Jr.



Latest Subscriber Content

[Embed Analysis: Which Embeds Do Clarion Developers Use The Most?](#)

Most Clarion developers spend most of their time writing embed code. But new (and sometimes experienced) developers are often faced with a bewildering question: Which embeds should I use, and why? Dave Harms provides some preliminary results on typical usage from the ClarionMag embed analysis project.

Posted Tuesday, January 30, 2007

[Clarion 101: Designing A Database](#)

The first step in almost any Clarion application is sorting out the underlying database structure. Dave Harms continues the 101 series with a new database for our old friend Bill Gates.

Posted Friday, January 26, 2007

[Accessing Private Class Data](#)

Ever get annoyed by private data in the ABC (or any other) classes? Jeff Slarve has a neat trick for bypassing the PRIVATE attribute.

Posted Friday, January 26, 2007

[.NET for Clarion Developers](#)

Dave Harms looks at the differences between Win32 and .NET development, and between writing Clarion 32 bit Windows apps and writing Clarion.NET apps.

Posted Friday, January 19, 2007

[Clarion Magazine Wants Your TXAs!](#)

Clarion Magazine is looking for TXAs from real-world applications for an upcoming Clarion 101 article on embed points. All TXAs will be kept confidential and will only be used to collect statistics on which embed points are most often used. To contribute a TXA just open your app, choose File|Export Text, and send the resulting TXA to ed@clarionmag.com. Please do not send apps or data dictionaries, just TXAs.

Posted Tuesday, January 16, 2007

[Signing Your Applications: New Challenges](#)

IE7 introduces some new security features which can cause warning messages to appear when users run your applications across a network. Jane Fleming revisits the topic of application signing and shows how to meet the new security requirements.

Posted Friday, January 12, 2007

[Review: Clarion Training on Demand - SQL Programming Series](#)

Russ Eggen reviews SoftVelocity's recently released Clarion Training on Demand (CTOD) Course VII - SQL Programming Series.

Posted Wednesday, January 10, 2007

[PDF for December 2006](#)

All articles for December 2006 in PDF format

Posted Tuesday, January 02, 2007

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

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:



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

[ProScan, ProImage Demos](#)

Demos of ProImage 1.6 standalone and integrated with ProScan 1.5 as a page editor are now available.

Posted Thursday, February 01, 2007

[ProImage/ProScan Wedding Sale Extended](#)

The ProScan/ProImage "Wedding Sale" celebrates the long awaited union of ProScan and ProImage; as this event was delayed the special bundle pricing has been extended until midnight Sunday (February 4, 2007). During that time you can save \$25 off ProImage or ProScan or save \$75 on the bundle price.

Posted Thursday, February 01, 2007

[ProImage 1.6 Released](#)

ProImage 1.6 has now been officially released and is available for immediate download. This release includes: Full compatibility and integration with ProScan; Mouse Resize of the Visual Target System; The FreeHand crop mode allows the user to draw out an area that they want to use from the source image; TrueView Image Protection adds automatically resampling on a crop resizes the output to prevent distortion of the image at all times; Smart Image Clipboard makes it easy to use the copy selection feature to copy small representations to paste into emails as well as full size representations to paste into other programs; 264 free high quality icons for ProImage from www.icons-icons.com, a subset for ProImage from the "Lights", "Brights" and "Roma" icons sets; Numerous small enhancements and a few minor bug fixes.

Posted Thursday, February 01, 2007

ProScan 1.5 Released

ProScan 1.5 has now been officially released and is available for immediate download. Features include: Full compatibility and integration with ProImage; TIFF Navigators that allow your users to "turn the page" of a TIFF displayed on a standard Browse or Form (without loading the Scan Manager); A Shuffle Feature that allows users to move a page forward or backwards in a TIFF document to correct when a page is scanned out of order; Backwards compatibility with Clarion 5.5 (to match that of ProImage); Custom compression settings for individual scanner profiles; Other numerous small enhancements and a few minor bug fixes.

Posted Thursday, February 01, 2007

Comment on Clarion News

You can express an opinion on the news on the ClarionLife news page.

Posted Thursday, February 01, 2007

xReportPreview 3.6

New in xReportPreview 3.6: Fix to bottom page header bug; Set left and right offset for headers; Option to start a preview window on the same thread.

Posted Monday, January 29, 2007

FullRecord 1.80 Beta

A beta of FullRecord 1.80 and available for those with a valid (current) maintenance pack. From February 1st on, FullRecord's price will be US\$ 129, and the Audit Pack will cost US\$ 149. Main new feature on version 1.80; "Field" inspector window that can be activated on request (with a button), or in the whole system with a global hot-key.

Posted Monday, January 29, 2007

BackItUp 1.9.0

BackItUp now comes as a SetupBuilder executable install.

Posted Monday, January 29, 2007

Rasp 0.2

Archer Technologies Rasp is an online file management system. This release adds complete control over File Types (eg. "Document") and their Extensions (eg. ".doc", ".txt", ..). The File Types are used to recognize and categorize files. The File Import window has been revamped, and includes support for automatic and manual import. You can also synchronize with the Rasp web server's directory structure.

Posted Monday, January 29, 2007

Clarion Desktop 2.61

Clarion Desktop 2.61 is now available. This version includes several significant improvements and fixes

Posted Monday, January 29, 2007

Free Vista Icons, Wallpaper

Every month, icons-icons.com gives away free icons and wallpaper to newsletter subscribers. This month's free icons are cyan colored and Vista-glassy looking, in shields and circles, with various center images. There are a total of 14 free icon images this month, each in four sizes and six graphic formats. This month's free wallpaper follows the cyan theme: it's a photo of icebergs on their way to sea. The wallpaper comes in a large variety of sizes, including super-wide for dual monitors, and tiny for handhelds. It's a free download for newsletter subscribers.

Posted Monday, January 29, 2007

New CircloCyan Icon Set

The new CircloCyan Icon Set from Icons-Icons includes 115 of the most popular icon images, including a "blank" for those who need it. This set has Vista's glassy look, and include Vista's new larger sizes. Nine sizes are included, from tiny 16x16 all the way up to 256x256 pixels, in six graphic formats. On sale for \$9.95 through January 31, with a money-back guarantee.

Posted Monday, January 29, 2007

BackItUp 1.8.0

BackItUp 1.8.0 includes some code changes and the option to generate an MD5 hashfile for each file added to the backup, and to manually generate a hash from any file. (MD5 and SHA256). Each download now also has the MD5 hash available so you can verify if a

file is correctly downloaded and unmodified. (Hover over the download to see the MD5 hashcode.) To verify, use one of the many MD5 hash generators out there or use BackItUp.

Posted Wednesday, January 24, 2007

[Clarion Daily News](#)

The ClarionLife Clarion Daily News page is now officially open.

Posted Wednesday, January 24, 2007

[Vivid Help Conductor Template Updated](#)

The Vivid Help Conductor template allows developers to write template style code without writing template or knowing anything about template language. For instance, you can use the Conductor template to place generated code in all instances of a certain embed point. This release adds access to report controls.

Posted Wednesday, January 24, 2007

[BackItUp Updated](#)

The latest release of BackItUp fixes a bug with backing up changed files only using the initiate backups setting. Also BackItUp is now Clarion Desktop compatible.

Posted Wednesday, January 24, 2007

[SetupBuilder 4.x Upgrade To SetupBuilder 6.0](#)

If you would like to upgrade your SetupBuilder 4.x license to SetupBuilder 6.0, please send an email to sales@lindersoft.com and learn more about your upgrade options.

Posted Wednesday, January 24, 2007

[Gitano Source Code Deal Ends This Week](#)

In January all Gitano Source Code Utilities are on sale for \$99 each. Products include gCal, gCalc, gNotes, gFileFind, gQ and gSec.

Posted Wednesday, January 24, 2007

[xNotes Browse-Form 1.3](#)

xNotes Browse-Form 1.3 is now available. This release adds support for MS SQL and an

MS SQL example.

Posted Wednesday, January 24, 2007

NeatMessage 2.10

NeatMessage 2.10 adds a Silent Mode feature to prevent messages from showing up.

Posted Wednesday, January 24, 2007

Advertising on Clarion Desktop!

There are now 383 Clarion developers using Clarion Desktop, which now offers advertising space to suppliers. For more info see www.clariondesktop.com and click on "Advertise" at the top of the page, or email Gary.

Posted Wednesday, January 24, 2007

SetupBuilder 6.0 Build 1750

Lindersoft has released SetupBuilder 6.0 Build 1750, a maintenance release. If you have a current SetupBuilder Maintenance and Support Subscription Plan, the update is free of charge. You can get the latest version by selecting "Check for Updates" from within the SetupBuilder 6.0 IDE.

Posted Wednesday, January 24, 2007

Next UKCUG Meeting

The next UKCUG is going to be on either Mon March 19th or Mon March 26th at New Hall, Cambridge University. As always you can vote on what you would like to form the agenda, and if you want to suggest a topic or even volunteer as a speaker then please feel free to get in touch.

Posted Wednesday, January 24, 2007

2007 Aussie DevCon and Capesoft Training

The 2007 Aussie DevCon and Capesoft Training will be held May 23-27, 2007, in Sydney at the Novotel Darling Harbour Hotel. The featured speaker is SoftVelocity president Bob Zaunere who will be imparting his vision for things Clarion. As part of the lead up to the Aussie DevCon, CapeSoft will be presenting three days of "hands on" workshop training in many of their products. There are two early bird specials. Each lasts one month only. Seating is strictly limited and is on a first come first served basis - (the only priority goes to

presenters - including local ones - wink wink say no more - apply now), so make sure you get in early. Special rates are available at three hotels, the Novotel Darling Harbour, The Ibis Darling Harbour, and The Grand Mercure Apartments . They are all in the same complex, with the Ibis being the cheapest and the apartments being good value for families. Please support these hotels if you can, as it is part of the conference package to fill a set number of rooms to get better conference rates. Remember to quote AUSSIE DEVCON to get the better rates. Overseas Attendees - check the web site for important immigration advice. Fees have been kept to a minimum, well under \$1000 for both events, that's below 50% of other similar events.

Posted Wednesday, January 24, 2007

Capesoft MailSync Announced

Capesoft MailSync (formerly Sterling Data's ExpressFlash) allows you to import mail, newsgroups, folders and contacts from Microsoft Outlook Express (which has been renamed to Windows Mail with the release of Windows Vista). In addition to support for Outlook Express, MailSync also supports limited functionality for Microsoft Outlook (like importing contacts). MailSync ships as source code and Clarion templates, there are no DLLs or black boxes. MailSync is backward compatible with ExpressFlash, and existing ExpressFlash users will be eligible for a free upgrade to MailSync (which has updated documentation and examples). Normally retailing for \$99, CapeSoft has a special Valentine's introductory offer (\$79) through 14 February 2007.

Posted Wednesday, January 24, 2007

CapeSoft World Tour 2007

The CapeSoft World Tour 2007 (aka "Round the world in 18 days") is now in the planning stages. Visits to Sydney, Australia; Las Vegas, USA; and Cambridge UK are all on the schedule for late May / Early June. Hint: This is nothing like the 2006 World Tour - see the web page for details. If you are in Australia then registrations are now open - There are discounts if you register early, so don't miss out. The first discount cut-off is Feb 15th. Tony York from the Australian Users Group has done a magnificent job negotiating with the hotel to keep rates really low for a really prime hotel.

Posted Tuesday, January 16, 2007

Clarion Desktop 2.50

Clarion Desktop 2.50 is now available. This version includes a new "Search" field (powered by Google) which you can use to search all the websites listed on Clarion Desktop simultaneously. As always, the standard edition is 100% free.

Posted Tuesday, January 16, 2007

Orlando Web Hosting

Long time Clarion user and Central Florida CUG president Chuck Simmerson has a new web hosting service. Plans available from \$6.50 per month.

Posted Tuesday, January 16, 2007

Huenuleufu Alternate URL

As a consequence of a recent newsgroup thread on name recognizability Jorge Lavera has adopted an alternate Hueluleufu URL. The original URL is still in use, and the company name will not change.

Posted Tuesday, January 16, 2007

CHT January Newsletter

The CHT January newsletter includes some interesting possibilities and speculation about using .NET libraries in C6/C7.

Posted Tuesday, January 16, 2007

Clarion Magazine

Clarion 101: Designing A Database

by Dave Harms

Published 2007-01-26

The Clarion development environment lets you build Windows business applications incredibly quickly which, naturally, is why it's taken a full six articles in this series to get to the point of actually creating an application. (Actually it's going to take a few more because first it's time to head back to the data dictionary.)

The reason for this lengthy road to rapid application development is not that you can't create an app within minutes of opening the box (you can – look in the Help for the Online User's Guide, Getting Started topic), it's because so much of Clarion's power is premised on you, the developer, doing your homework *before* Clarion whips up an application on your behalf.

I want to be clear about how Clarion accomplishes its RAD wizardry. And much of that wizardry depends on what's in your data dictionary.

In brief, application development in Clarion typically looks like this:

1. Design the database (if the database already exists you can skip ahead to any of steps 2-5 as appropriate))
2. Import or manually enter the tables in the data dictionary
3. Define relationships between tables, if not already present
4. Define data integrity rules, if not already present
5. Generate one or more applications using the application wizard
6. Customize the application(s)

There are several common variations on this development path, including (but not limited

to) hand coded apps, dictionary-free apps, and heavily wizarded apps.

Hand coding

Not every Clarion application depends on the data dictionary and the AppGen. It's quite possible, and sometimes desirable, to write very small programs entirely by hand. Hand coding large applications used to be more common simply because the application generation environment, in the early years (I'm thinking particularly of Clarion Professional Developer for DOS), often lacked needed functionality and couldn't easily be modified to produce highly customized programs. But there are few excuses now to do without the data dictionary and AppGen.

Dictionary-free apps

Just because you use the AppGen doesn't mean you have to use a data dictionary. You can still create an application with menus, windows, controls etc. although you'll generally have to build your application up one window at a time rather than using the wizards.

Heavily wizarded apps

As you'll see in the next article, Clarion's wizard technology makes it easy to create a basic, functioning application out of a data dictionary. Clarion's wizards (or, more accurately, wizatrions) are highly configurable and greatly underused tools. But some developers have figured out that they can save themselves a lot of work by doing more work up front on the data dictionary and the wizard settings. They'll create data dictionary, wizard an application, evaluate the app, make further changes to the dictionary and/or the wizard settings, and wizard up a new app. Repeat as needed. Only once they've wrung everything they can out of the wizards and the dictionary do they begin modifying the application.

The mainstream approach

In this series I'll concentrate on the mainstream approach to Clarion development, which involves the five steps listed above, executed in sequence, with perhaps a couple of

iterations of the #3-#5 loop. Over the course of development the vast majority of time is spent in step 6, but careful work in the previous steps will also save a whole lot of effort once you get to that last step.

Beginning the data dictionary

It's finally time to start working on your application, and in the data-intensive world of Clarion development that means getting the data dictionary together. How you do this depends largely on whether or not you have an existing database.

If you have an existing database then most likely you'll be able to import it into the data dictionary. You can import tables one at a time using the File|Import Table option or, if you have Enterprise Edition, you can use Multi-Table Import which employs the Dictionary Synchronizer.

If you're developing from scratch, you'll want to know a little bit about database design. If you haven't already done so, I recommend you read Bruce Johnson's two articles in this series, [Describing your Data](#) and [Understanding Keys and Indexes](#). I also suggest you go through Tom Ruby's excellent [Five Rules for Managing Complexity](#), a series on normalizing databases.

The scenario

Let's begin with one of those highly plausible scenarios common in articles of this type. Suppose you're a close personal friend of Bill Gates, and ol' Bill has been bitten hard by the green bug. He's replaced all the incandescent lighting in his Seattle mansion with compact fluorescent bulbs, and he wants to track exactly how much energy those new bulbs are using. He already has a data feed with the following information:

LocationID	Integer ! Location of each light bulb
TimeOn	DateTime
TimeOff	DateTime

What Bill wants is a set of reports showing how much energy he's saving over the equivalent incandescent lighting, and he wants to know that by individual bulb, by room, by day of the week, by manufacturer, and probably a few dozen other ways. The problem

is his VB.NET programmers have been working on the problem for months already and still don't have a useful product. But Bill knows you know Clarion, and he has a gut feeling you can help.

The database

Databases are almost always more complex than they seem at first. Consider the miserly three fields of information in Bill's data feed. Can you really generate all the reports Bill wants just from that data? Clearly you need some additional information. At a minimum you'll need to know something about:

- the rooms in Bill's mansion (so you can identify a bulb with a room)
- the rated energy usage of the various bulbs
- the time a given bulb is on (this you get from the data feed)

So what are the very obvious entities you can pull out of this information? Clearly you'll need tables for:

- the list of rooms
- the data feed showing when bulbs are turned on/off

and there's a good chance you'll also want:

- a list of bulb brands/manufacturers

How about the light bulbs themselves? Here's where things get a little interesting. You may be tempted to create a `Bulb` table with the following information:

<code>BulbID</code>	<code>Integer</code>
<code>MfgrID</code>	<code>Integer</code>
<code>LocationID</code>	<code>Integer</code>
<code>Wattage</code>	<code>Integer</code>

A `Bulb` record like that would certainly contain all of the necessary information, but it would also be duplicating some information about the bulb. Although it's possible Bill would want a unique type of light bulb in each socket, it's more likely he'd be using many of the same type of bulb. Accordingly, you'd want to keep a table with `BulbType` information, linked in a many-to-one relationship to the manufacturer table.

You can begin entering these tables directly into the data dictionary, but I'm going to take the more traditional route of diagramming the database first. Clarion 7 will reportedly have a visual diagramming tool, but since I don't have C7 yet I'll demonstrate the diagram using [Datanamic's Dezin for Databases](#) (version 3). Figure 1 shows the database diagram so far.

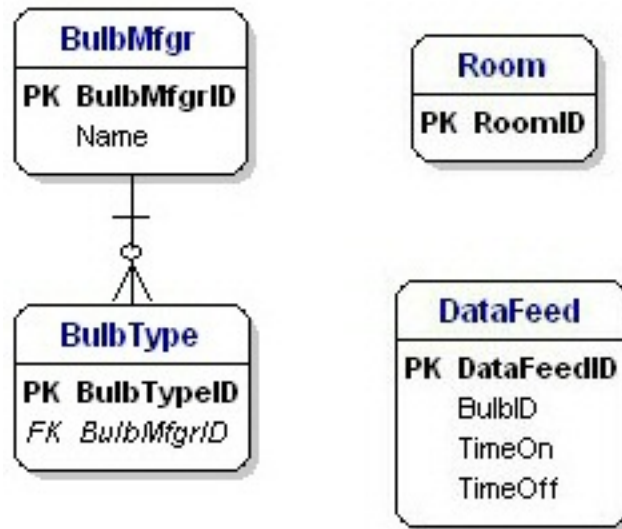


Figure 1. Preliminary data diagram

Take note of the naming convention I'm using in Figure 1. I name all primary key fields as the name of the table with the suffix 'ID'. This consistency makes it easy to deduce the primary key from the table name and vice versa. Similarly, I always name a foreign key field the same as the primary key field to which it relates. That makes it easy to keep track of relationships, and it also enables the dictionary editor to match fields up by name when you're defining relationships.

You've defined the first relationship, between `BulbType` and `BulbMfgr`, and that's a good start. But how do you get from `BulbType` to the data feed? The answer depends on what kind of information you want to track.

You already know that Bill wants usage information not just by room but also by bulb. You'll need some sort of record to represent bulb location, so you set up a `Socket` table:

<code>SocketID</code>	Integer
<code>RoomID</code>	Integer
<code>Description</code>	Text

Now you're ready to think about linking a bulb type to a socket. Each socket will have a single bulb type, right? Well, only partly true. At any given moment in time that socket will have a single bulb type, but quite possibly the bulb type will change over time since bulbs periodically need to be replaced, or may be superseded by a more efficient model. This changes the relationship in a subtle but important way. Instead of a one (bulb type) to many (sockets) relationship you now have a many to many relationship, since you can also have more than one bulb type per socket. And that means you need a linking table called `BulbTypeSocket` to represent each actual bulb:

<code>BulbTypeSocketID</code>	Integer
<code>SocketID</code>	Integer
<code>BulbTypeID</code>	Integer
<code>StartDate</code>	DateTime

Figure 2 shows the database with this table and the remaining relationships filled in.

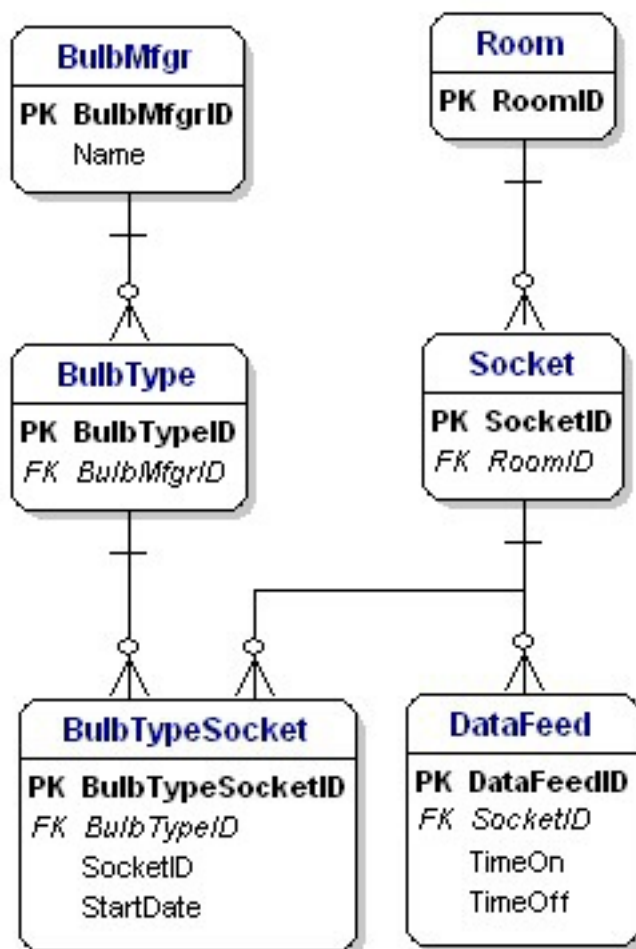


Figure 2. Data diagram nearly complete

You're feeling pretty good about this database design, and you're sitting on Bill's lakeside patio sipping an iced cappuccino when Bill casually mentions he's really looking forward to the breakdown on the net cost per bulb per year on a shipment of new compact fluorescents he's just received. And it hits you: you have no way of tracking individual bulb cost in your database.

The problem is you've only been thinking about energy consumption as determined by the data feed, and you've ignored the capital cost of the bulbs. So clearly storing the `BulbTypeID` in the `BulbTypeSocket` record is too abstract: you'll need a record for each actual, physical bulb so you can track the purchase cost of that bulb. The answer is to add a `Bulb` table between the `BulbType` and `BulbTypeSocket` tables. You still want a `BulbType` record so you don't duplicate bulb information such as wattage (and perhaps expected life). But `BulbTypeSocket` really now relates to `Bulb` and `Socket`, so change its name to `BulbSocket`. Don't forget to change the primary key name as well. Figure 3 shows the result.

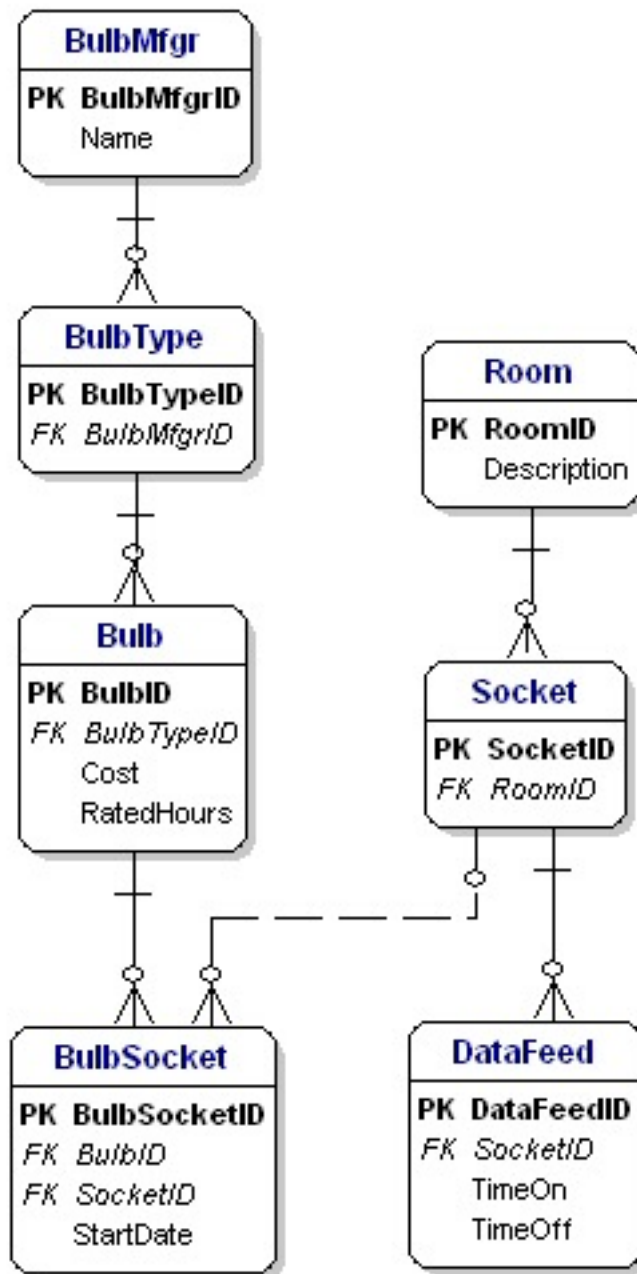


Figure 3. Data diagram with Bulb table

One small point is that the `BulbSocket` record has only a `StartDate` (datetime) field, and no `EndDate`. You might make this design choice on the assumption that the `EndDate` of one record is the `StartDate` of any following record, in which case an explicit `EndDate` is duplicate data. That's fine for this application, but there may be circumstances in other applications where an assumption like this is not valid, and where an `EndDate` must be specified. Oftentimes database design involves a tradeoff between convenience and precision; just make sure you're precise where it really counts! And some tasks are made easier by a little duplicated data, so it may yet become apparent that it's worthwhile having an `EndDate` in this table.

As you can see by this example, the more you think out your requirements during the database design stage the less work you'll have to do when it comes time to develop the application. Deciding to track bulb costs resulted in a significant change to the database, but it's trivial to make the change now; it's not so easy once you have a bunch of browses, forms and reports all of which are built on the notion of a `BulbTypeSocket` table linked to `BulbType` and `Socket`. Consider every possible output (e.g. report) and evaluate your database. Can you get the data you need from your database design?

Can you think of any other tables you'd want to add? The `Bulb` table has a cost field, but Bill isn't going to buy bulbs one at a time. It's more likely he'll buy them by the box (or pallet) and the per-bulb cost will be calculated. You could add a whole inventory management module and a bunch more tables, but for the sake of brevity I'll assume that information is tracked elsewhere, in an existing system, and the per-bulb cost will be magically added to this system when the time comes.

The data diagram now shows seven tables, which seems like a lot for a simple task like calculating lighting energy efficiency! If you've done a good job of assessing the inputs and outputs you probably have just about everything you need. I'm a big fan of modest over-engineering at design time; when a client asks for a new feature, it's always best to have a database design that already accommodates the request.

In the next installment in this series I'll plug this database design into the data dictionary and begin talking about relationships and data validation.

[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 is a member of the American Society of Journalists and Authors ([ASJA](#)).

Reader Comments

[Add a comment](#)

- [» Dave, As always your articles are always very...](#)
- [» Thanks, Jim! Dave](#)

Clarion Magazine

Embed Analysis: Which Embeds Do Clarion Developers Use The Most?

by Dave Harms

Published 2007-01-30

Most Clarion developers spend most of their time writing embed code. But new (and sometimes experienced) developers are often faced with a bewildering question: Which embeds should I use, and why?

In an effort to answer this question I recently asked ClarionMag readers to send me their TXAs so I could compile some data on embed usage. I've received over twenty TXAs so far, enough to prime the pump. In this article I'll offer a preliminary look at the data, including the top seven embeds. In a later article I'll draw more detailed conclusions about which embeds should be at the top of your list.

Methodology

I'm currently using a three-table structure to store the embed information in a MySQL database. The EmbedApp table simply stores the TXA name; EmbedProc stores the procedure name; and Embed stores the embed information.

Here are the table structures:

```
EmbedApp          FILE, DRIVER( 'ODBC' ), OWNER( glo:owner ), |
                  NAME( 'EmbedApp' ), PRE( EMA ), BINDABLE, THREAD
PRIMARY          KEY( EMA:EmbedAppID ), PRIMARY
Record           RECORD, PRE( )
EmbedAppID       LONG
TXA              CSTRING( 61 )
                  END
                  END
```



```

EmbedProc          FILE, DRIVER( ' ODBC ' ), OWNER( glo:owner ), |
                   NAME( ' EmbedProc ' ), PRE( EMP ), BINDABLE, THREAD
PRIMARY           KEY( EMP: EmbedProcID ), PRIMARY
EmbedAppIDIndex   KEY( EMP: EmbedAppID ), DUP
Record            RECORD, PRE( )
EmbedProcID       LONG
EmbedAppID        LONG
Proc              CSTRING( 61 )
ProcFromABC       CSTRING( 61 )
ProcCategory      CSTRING( 61 )
                   END
                   END

Embed              FILE, DRIVER( ' ODBC ' ), OWNER( glo:owner ), |
                   NAME( ' Embed ' ), PRE( EMB ), BINDABLE, THREAD
PRIMARY           KEY( EMB: EmbedID ), PRIMARY
EmbedProcIDIndex  KEY( EMB: EmbedProcID ), DUP
Contents          BLOB
Record            RECORD, PRE( )
EmbedID           LONG
EmbedProcID       LONG
LinesOfCode       LONG
Embed             CSTRING( 61 )
Param1            CSTRING( 61 )
Param2            CSTRING( 61 )
Param3            CSTRING( 61 )
Priority           LONG
                   END
                   END

```

Storing the embed information is fairly straightforward, but getting the information in the first place is a little more complicated. One way is to use an embed-extraction template, and this would certainly have been the easiest way to get the information in a form ready to be inserted into my database. On the other hand if I took that approach anyone who wished to contribute would have to register and run that template, and not everyone is happy about having to register yet another template. To make it as easy as possible for readers to participate, I elected to ask for TXAs and then parse those files looking for the embed data.

That may have been a mistake.

A TXA is simply a text version of your application, one which can be reconstituted as an APP as required. It's also possible to export just specific procedures/modules in a TXA. In some future perfect world, Clarion TXAs will be XML files, and when I want to do something like this I'll simply query them with Clarion.NET using LINQ. But that is then; this is now. And TXAs are a bear to parse.

I've written some parsing code, using the [Finite State Machine](#) approach ably explained by John Christ. I really can't think of another good way to approach a task like this. I've also used Skip Williams' [Debugger](#) (yes, it is spelled that way) extensively, together with [DebugView](#), while testing my parsing code. I'm still working out a few issues, and I've learned more about TXA structures than I really care to know.

In any case, I now have a version of my TXA importer that does a reasonable, if not bulletproof, job of importing TXAs. What follows is a preliminary analysis of the data. I'll be expanding on this in a later article, and in the meantime I can always use more TXAs.

The stats

As of this writing I have 24 TXAs for a total of 966 procedures and 3972 embed points. Count by procedure type are as follows:

Count	Procedure Type
2	Splash
14	Frame
117	Source
145	Process
172	Report
510	Window

The Window procedure types break down as follows:

Count	Window Type
1	Selects
19	Window
56	not specified
189	Form
245	Browse

In general, procedure types seem relatively evenly distributed among Source, Process, Report, Browse, and Form, with browses being the most common.

The top-level embed list, from most- to least-used, is as follows, but don't jump to any conclusions just yet!

Count	Embed
1016	%WindowManagerMethodCodeSection
831	%ControlEventHandling
352	%BrowserMethodCodeSection
244	%ProcessManagerMethodCodeSection
209	%ProcedureRoutines
121	%ControlPostEventHandling
117	%DataSection
110	%TreeSectionMethodCodeSection
97	%WindowEventHandling
91	%NewMethodCodeSection
79	%ProcessedCode
56	%ControlHandling
56	%UltraTreeMethodCodeSection
41	%EditInPlaceManagerMethodCodeSection
34	%UTVMMethodCodeSection
29	%BreakManagerManagerMethodCodeSectionLevelAction
29	%ProcessActivity
24	%BeforePrint
23	%XPTaskPanelTaskClickedAfterCode
19	%LocalDataAfterClasses
19	%ControlPreEventHandling

17	%AfterFileOpen
17	%XPTaskPanelTaskLogicAfterCode
17	%BeforeFileClose
15	%LocalProcedures
14	%BeforeFileOpen
13	%BrowserEIPManagerMethodCodeSection
13	%ProcedureSetup
12	%DataSectionAfterWindow
11	%AfterFileDeclarations
10	%FileDropMethodCodeSection
10	%FileLookupMethodCodeSection
10	%BrowserMethodDataSection
10	%AfterPrint
9	%NewMethodDataSection
9	%FormatBrowse
9	%AfterWindowOpening
9	%DataSectionBeforeWindow
9	%FileManagerCodeSection
9	%GlobalMap
8	%AfterGlobalIncludes
8	%NetTalkMethodCodeSection
7	%ProgramSetup
6	%WindowManagerMethodDataSection
5	%TreeSectionMethodDataSection
4	%ProcedureInitialize
4	%UltraTreeMethodDataSection

4	%BeforeGlobalIncludes
4	%BeforeWindowOpening
4	%BrowseAfterChange
3	%FileLookupMethodDataSection
3	%PreviewerManagerMethodCodeSection
3	%PostWindowEventHandling
3	%BrowseBeforeInsert
3	%ModuleDataSection
3	%TEDropIDOk
3	%BrowseBeforeChange
3	%ProcessManagerMethodDataSection
3	%CustomAlertEmbed
3	%TEBrowseDropHandlingAfter
3	%ControlPostEventCaseHandling
3	%LocatorMethodCodeSection
3	%XPTaskPanelTaskLogicBeforeCode
2	%BrowseAfterDelete
2	%FM2EndLoopFM2InitCode
2	%TETreeDropHandlingAfter
2	%ProgramEnd
2	%AfterFileClose
2	%ResizerMethodCodeSection
2	%AcceptLoopBeforeEventHandling
2	%ProgramProcedures
2	%GlobalData
2	%ReportTargetMethodCodeSection

2	%AdditionalDebugHooks
1	%ErrorManagerCodeSection
1	%DasTagAfterTagOnOff
1	%BrowseBeforeDelete
1	%AfterDctInitialization
1	%AfterEntryPointCodeStatement
1	%DasTagAfterUnTagAll
1	%AnyFontABCEnable
1	%BeforeWindowMakeover
1	%AfterClosingExports
1	%BeforeWindowClosing
1	%BeforeAddingStyles
1	%RelationManagerCodeSection
1	%DasTagBeforeTagAll
1	%ChildViewManagerMethodCodeSection
1	%BeforeFileDeclarations
1	%ErrorManagerDataSection
1	%DasTagBeforeTagOnOff
1	%ListboxStyleAfterDefine
1	%BrowsePrepNormal
1	%FieldLevelValidation
1	%FileDropComboMethodCodeSection
1	%BrowseAfterInsert
1	%StartOfModule
1	%NetTalkRefreshCode
1	%AnyFontABCDisable

1	%BrowseBoxAfterUpdate
1	%AfterDctDestruction
1	%dMethodCodeSection
1	%DasTagBeforeUnTagAll
1	%DataSectionBeforeReport
1	%ProgramRoutines
1	%AfterTurnQuickScanOn
1	%RelationManagerDataSection
1	%DasTagAfterTagAll
1	%FEPreCodeSection
1	%PrimeFields
1	%BrowsePrepSelectRecord

As you might have guessed already, some of those embed points don't refer to just one embed; instead, they refer to a section of embeds. Here's the breakdown on %WindowManagerMethodCodeSection by method name:

Count	WindowManager Method
668	Init
23	Kill
22	Reset
20	OpenReport
20	PrimeFields
20	TakeCompleted
17	SetControlProperties
15	Open
14	PrimeUpdate
13	InitControlProperties

10	AskPreview
9	TakeNoRecords
5	InsertAction
4	TakeWindowEvent
4	EndReport
4	TakeRecord
3	TakeFieldEvent
3	TakeSelected
3	Run
3	Update
2	TakeEvent
2	TakeAccepted
2	SetControlValues
1	TakeCloseEvent
1	Ask

The data for %ControlEventHandlering looks similar, although instead of methods you'll see various field equates.

Here's a list of control embed points by event name:

Count	Event
441	Accepted
25	NewSelection
9	Drop
5	Selected
4	TabChanging
3	AlertKey

2	Drag
---	------

It's not too surprising that the vast majority of embed code for controls executes when a control's value is accepted.

Stripping out the control-specific embeds yields the following overall list of the most commonly-used embed points:

Count	Embed	Parameter
532	%WindowManagerMethodCodeSection	Init
176	%ProcessManagerMethodCodeSection	TakeRecord
151	%ProcedureRoutines	
98	%DataSection	
79	%ProcessedCode	(source procedure)
48	%BrowserMethodCodeSection	SetQueueRecord
42	%NewMethodCodeSection	
40	%ControlPostEventHandling	various controls
33	%BrowserMethodCodeSection	ValidateRecord
29	%ProcessActivity	
27	%WindowEventHandling	OpenWindow
27	%NewMethodCodeSection	
26	%ProcedureRoutines	Init
26	%BrowserMethodCodeSection	
24	%BeforePrint	
20	%WindowEventHandling	Timer
18	%BrowserMethodCodeSection	
16	%AfterFileOpen	TakeNewSelection
15	%BeforeFileClose	
15	%ProcessManagerMethodCodeSection	Kill

15	%ProcessManagerMethodCodeSection	Kill
15	%ControlHandling	
15	%WindowEventHandling	
14	%WindowManagerMethodCodeSection	Open
13	%BreakManagerManagerMethodCodeSectionLevelAction	TakeEnd
13	%WindowManagerMethodCodeSection	Kill
13	%WindowManagerMethodCodeSection	Reset
13	%ProcedureRoutines	TakeRecord
13	%BrowserMethodCodeSection	ApplyFilter
13	%WindowManagerMethodCodeSection	TakeCompleted
13	%BeforeFileOpen	
12	%WindowManagerMethodCodeSection	PrimeFields
11	%WindowManagerMethodCodeSection	OpenReport
11	%WindowEventHandling	GainFocus
11	%AfterFileDeclarations	
10	%WindowManagerMethodCodeSection	SetControlProperties
10	%WindowManagerMethodCodeSection	Kill
10	%AfterPrint	
10	%WindowEventHandling	CloseWindow
9	%WindowManagerMethodCodeSection	Reset
9	%BreakManagerManagerMethodCodeSectionLevelAction	TakeStart
9	%WindowManagerMethodCodeSection	OpenReport

Clearly the `WindowManager . Init` method is the big dog in Clarion embeds; you put code into this method more often than any other. Actually `WindowManager . Init` embodies a whole set of embeds: here is the breakdown of priorities for that method:

Count	Priority
6	300
2	450
2	500
19	501
1	1000
1	1300
1	1500
2	2250
1	2300
10	2500
2	2800
3	3001
1	3600
1	3700
3	3800
1	4000
19	4050
21	4500
2	4501
10	4901
3	4950
5	5000
25	5001
3	5050
2	5101

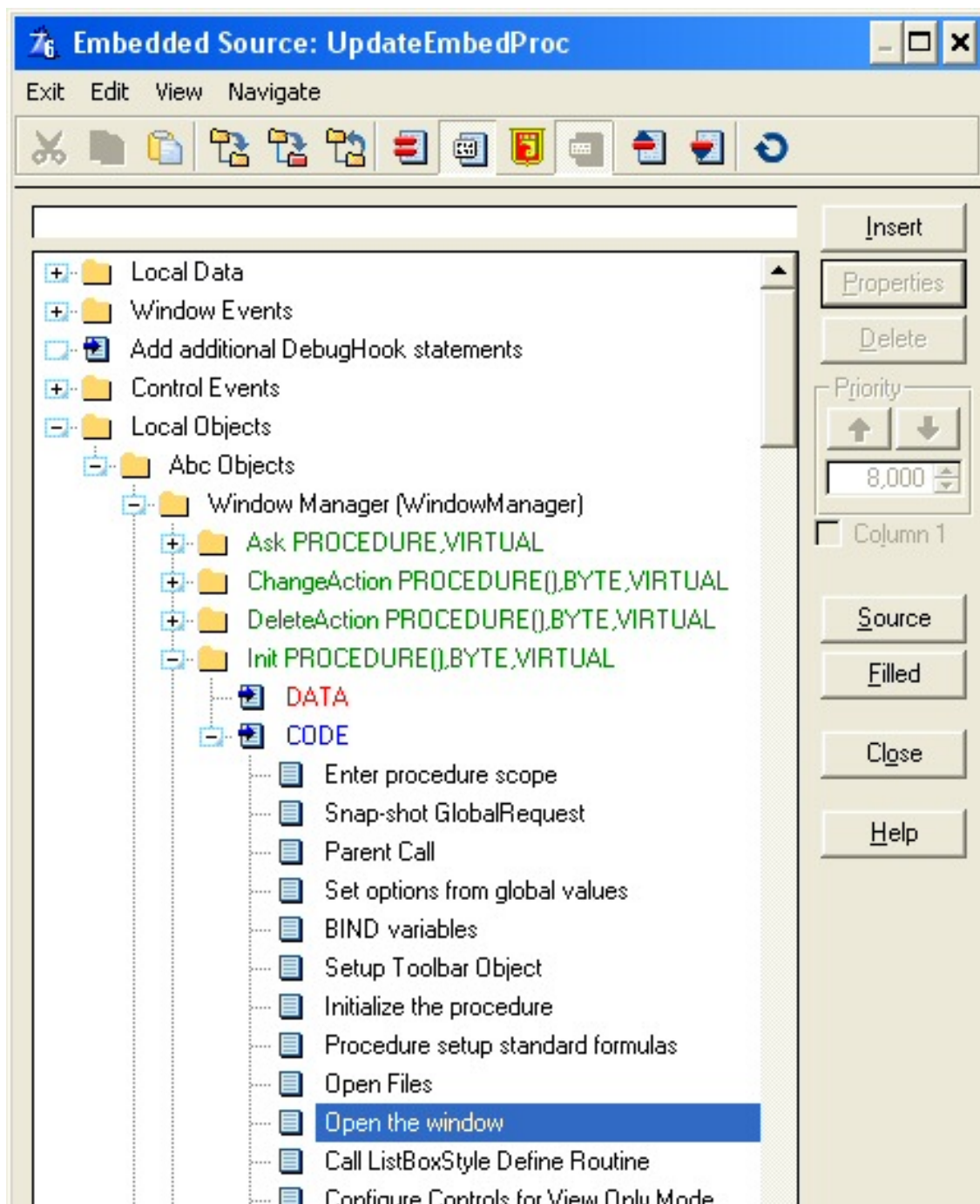
4	5300
1	5501
2	5600
1	5800
1	6000
58	6001
6	6300
2	6400
1	6500
2	7000
11	7001
11	7300
6	7500
99	7501
1	7550
3	7600
3	7700
1	7750
2	7751
32	7800
7	7900
6	8000
72	8001
1	8010
2	8020
14	8030

7	8050
2	8051
1	8053
1	8080
1	8090
3	8126
1	8140
1	8150
2	8180
1	8200
1	8250
4	8251
2	8330
1	8376
4	8450
2	8500
6	8501
4	8505
9	8550
5	8800
5	9000
97	9001
1	9300
6	9500
1	9600
2	9700

3	9800
---	------

To see where these embeds fit in the grand scheme of what the `Init` method does, go to the embed list for any `Frame`, `Browse`, or `Form` and look in the embed list for the `WindowManager` (see Figure 1). Navigate down the tree until you find the `Init` method. As you highlight each method you can see the corresponding priority level in the grayed-out spin box on the right hand side. (When you insert a code template or source the priority spin box becomes active.)

The most often-used `WindowManager.Init` embeds are on opening the files, opening the window, and preparing the alert keys. Alert keys has become a very common place for window setup code since it comes at the end of the `Init` method, after all the standard setup code has executed.



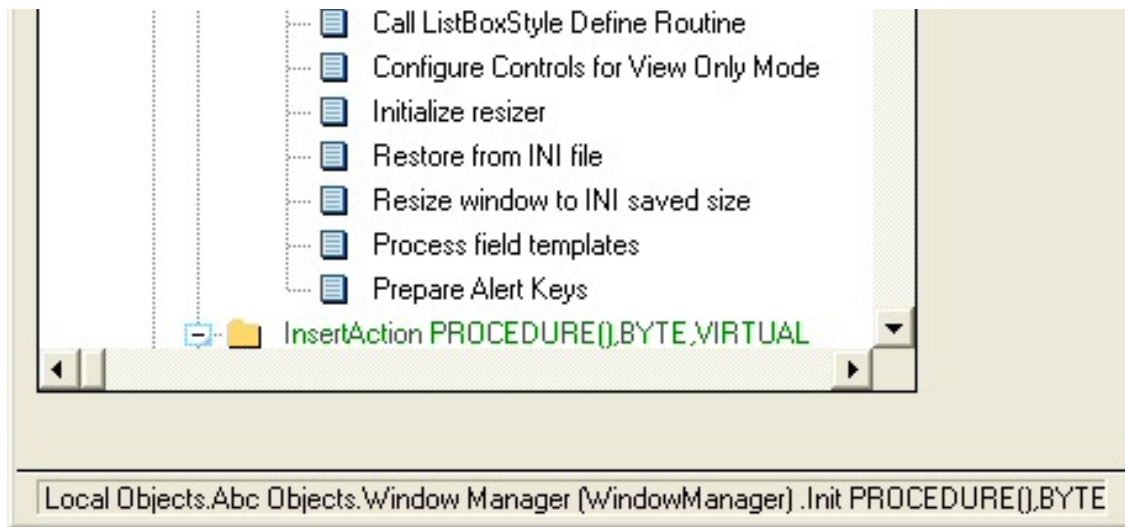


Figure 1. The embed list

The top seven

Based on the preliminary data, here are the top seven embed points.

1. `WindowManager.Init` method (as discussed)
2. Control handling – these are the embed points that show up when you double-click on a control in the window formatter.
3. `TakeRecord` method– in reports and processes, this method is called each time a record is retrieved. You might use this embed point to look up related data (if that can't be done conveniently in the templates), update some data, calculate a local variable, or the like.
4. Procedure routines – a place to put your local routines.
5. Local data – hand-coded declarations of local variables.
6. Code embed – for template-based source procedures.
7. `BrowseClass.SetQueueRecord` – copies the data from the table (file) to the queue used by the browse. This method is called for each record retrieved from the database. Among other things, you can use this embed point to create calculated fields for the browse.

ClarionMag still wants your TXAs

As I mentioned earlier I'll have a more detailed analysis of the data in a later article. Meanwhile I'm still looking for more TXAs, so if you have one to spare please [email it to me](#). Just send the TXA, not the DCT or the APP. I will keep the contents of your TXA confidential, and will only publish analysis, not actual embed contents.

My thanks to the TXA contributors to date! They include:

- Jim Halpin
- Lynn Howard
- Des Yaxley
- Andrew Crockett
- James Lishman
- Vido Vouk
- Bill Wilson
- Cliff Cady
- Jorge Herrera

[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 is a member of the American Society of Journalists and Authors ([ASJA](#)).

Reader Comments

[Add a comment](#)

Clarion Magazine

Accessing Private Class Data

by Dave Harms

Published 2007-01-26

Here's a little something from Jeffy's bag of hacks. Occasionally you may find yourself wanting access to a class's private variable, usually somewhere deep in the bowels of ABC.

Now, private variables are usually private for one of several reasons: a) giving a private variable a new value at an unexpected time could cause an error or even a GPF, b) the developer may have plans to remove or change the variable during a rewrite, or c) there's simply no plausible excuse for anyone to see it or change it. So if you do muck about with something the developer intended to be private, you're on your own. No warranty express or implied.

The usual way to get at something that's private in an ABC class is to remove the `PRIVATE` modifier in the declaration. That works, although you'll need to reapply the change when you next update your Clarion installation.

Jeff Slarve's approach doesn't require you to change any source code. Instead, it uses `WHO` and `WHAT` to examine the class's structure, find the matching variable, and return a reference to that variable. And since your code declares that reference, you can do whatever you like with it.

Nifty.

Just don't tell anyone you're doing this.

Here's the source to a short program that includes Jeff's FindWhere function:

Program

```

    map
      FindWhere(*Group pG,String pLabel),Long
    end

Include('Abreport.inc'),ONCE

R          ReportManager
OpenFailedLocation Long
MyOpenFailed      ANY
Code

OpenFailedLocation = FindWhere(R,'OpenFailed')
If OpenFailedLocation
  MyOpenFailed &= WHAT(R,OpenFailedLocation)
  MyOpenFailed = 100
  !R.OpenFailed variable in the debugger is 100
  message('MyOpenFailed=' & MyOpenFailed |
          & '||The position of OpenFailedLocation is ' |
          & OpenFailedLocation)
else
  Message('Could not find the "OpenFailed" property','Darn')
end

FindWhere Procedure(*Group pG,String pLabel)!,Long
MAX_LABEL_SIZE EQUATE(300)! Not sure what the
                          ! maximum size of a label is

A      ANY
Ndx    Long
Label  String(MAX_LABEL_SIZE)
Code

Ndx = 0
Loop
  Ndx += 1
  Label = WHO(pG,Ndx) ! Check to see if there's a label
  ! If there isn't a label, that doesn't necessarily
  ! mean that it's the end of the group. It's
  ! possible that there are some label-less properties.
  If NOT Label
    A &= WHAT(pG,Ndx) !Attempt to get a reference

```

```

    If A &= NULL          !If that's not successful
        Break            !Give up
    end
    A &= NULL
end
If Upper(Label) = Upper(pLabel) ! If a match
    Return Ndx              ! Here's the position
end
end
Return 0

```

Jeff points out that you should check the return value from `FindWhere` for zero before assigning the reference.

The downloadable zip includes a project (PRJ) file and the source shown here. If you use this with your own project remember to add the following to your project defines:

```

_ABCD11Mode_=>0
_ABCLinkMode_=>1

```

Without those project defines the test program will GPF. If you implement `FindWhere` as a function in your APP the defines will already be set for you as appropriate.

[Download the source](#)

[David Harms](#) is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors ([ASJA](#)).

Reader Comments

[Add a comment](#)

- [» Very nice! Now for a way to access complex private class...](#)
- [» Hi John - This seems to work. I have no idea how...](#)

- [» Interesting! I'm going to try this in one of my classes...](#)
- [» So far your solution for accessing complex private class...](#)

Clarion Magazine

.NET for Clarion Developers

by Dave Harms

Published 2007-01-19

The careful reader will have noticed that the title of this article is ".NET for Clarion Developers" and not "Clarion.NET for Developers". I've previously written about basic .NET concepts (see [What Is .NET And Why Should I Care? Part 1](#) and [Part 2](#)) and you can be sure that in the future Clarion Magazine will have detailed coverage of Clarion.NET. In between .NET and Clarion.NET is a wide area of speculation, some of which has been addressed (and fueled) by SoftVelocity's occasional release of new information.

In this article I'll look specifically at the differences between Win32 and .NET development and, where possible, between writing Clarion 32 bit Windows apps and writing Clarion.NET apps. These differences include:

- Platform vs. operating system
- Deployment
- Managed vs. unmanaged code
- Internal app structure
- Windowing libraries
- Database access
- Object orientation
- Using variables
- Application design conventions
- Web development
- Clarion language changes
- Mobile device support

It's a platform, after all

Conceptually the biggest difference between .NET and Win32 is the former is a complete platform, while the latter is an operating system with an API to some standard libraries. In Clarion we currently create Windows executable files which usually depend heavily on the Clarion runtime DLL and, indirectly at least, a variety of calls to the Windows core libraries (via the Windows API). But other than those dependencies Clarion EXEs (and DLLs and LIBs) are self-contained. Even the internal operation of the application can have its own quirks, such as a register-based parameter passing scheme that's incompatible with most other languages.

.NET, on the other hand, is both a massive class library and clearly defined way of creating and managing applications designed to accommodate many different kinds of programming languages. .NET has a whole lot to say about how your application will be structured,.

Downside: Older systems may not have .NET installed, necessitating a big download. Really old systems may not be able to run .NET at all, or not adequately. Check out the [system requirements page](#) for more info.

Upside: Massive class library eases workload for developers and language vendors, and is more organized than Win32. Managed code is less likely to blow up (at least in theory). You can easily create apps with a mix of languages.

Deployment

Installing a Windows application with shared DLLs can be problematic; installing COM components can be frightful. Just ask anyone who's been to "DLL Hell." For the most part .NET does away with the whole business of having to register components because .NET classes expose information about themselves, and that information can be obtained at runtime by other processes. You can actually install and/or move .NET programs around using XCOPY. But as [this article on MSDN](#) points out, you may want to use an installer for:

- Installation, repair, or removal of assemblies in the global assembly cache.
- Installation, repair, or removal of assemblies in user-defined locations.
- Rollback of unsuccessful installations, repairs, or removals of assemblies.
- Install-on-demand of assemblies in the global assembly cache.

- Install-on-demand of assemblies in user-defined locations.

LinderSoft's [SetupBuilder](#) handles all of these tasks nicely.

Downside: You can still use COM with .NET so you may be stuck with some of the same install hassles anyway.

Upside: Deployment is a lot less fraught with peril when it comes to shared components.

Managed vs. unmanaged code

Although "managed" code may be relatively new to many Windows developers, Clarion developers have enjoyed a form of this since the beginning. The Clarion runtime (for the most part) saves you from memory overwriting errors and you don't have to worry about cleaning up most variable declarations, as memory allocated is automatically deallocated when the variable is no longer used. Until Clarion 4 and the NEW and DISPOSE operators you couldn't even allocate memory directly; you had to use the Windows API to do that. Clarion also enforces type safety and handles automatic type conversion, and can trap array index errors if you wish.

.NET's idea of managed code is a bit more sophisticated, but covers essentially the same points: automatic garbage collection, type safety, and array index checking. .NET also has fairly comprehensive security provisions, allowing administrators to restrict applications from certain operations.

Downside: You can mix Win32 code with .NET code using the interop layer (and vice versa – see [Wade Hatler's excellent series on mixing Clarion and .NET](#)), which means that some code will be unmanaged anyway. For instance, in late 2005 Bob Z [stated](#) that using the TPS and Clarion drivers would use interop code (built into the runtime library) to execute unmanaged Win32 code. (He also indicated these drivers might at some time be ported to managed code.)

In Clarion.NET using some drivers (such as TPS) will probably involve unmanaged code.

Upside: Similarity to Clarion means a shorter learning curve.

Internal app structure

.NET can be compiled into EXEs and DLLs which are loosely analogous to Win32 EXEs and DLLs except that instead of platform-specific (i.e. x86) binary code they contain IL (for Intermediate Language) code. At runtime the IL code is compiled into platform-specific binary code and executed.

But this idea of DLLs and EXEs is potentially confusing because .NET packages these files as assemblies. An assembly may contain just one file, or it may contain multiple files. And it has the extension DLL or EXE. You could have an assembly with an EXE extension and containing not just the EXE but also several DLLs, some image files, and a bunch of other stuff.

Downside: Assemblies add another level of complexity for the developer. More importantly, since the code in an assembly is IL code, it is relatively easy to disassemble, much more so than binary code. There are a number of obfuscation tools available to help conceal the inner workings of .NET assemblies.

Upside: It's handy to be able to bundle multiple files as one unit.

Windowing libraries

Clarion developers are accustomed to using a standard set of window controls (widgets) along with Clarion's ACCEPT-driven event processing model to build menus, browses, and forms. Many of these controls (like the list box) are not native Windows controls, but Clarion-specific custom controls. And these custom controls were created to get around the limitations of the standard Windows controls.

.NET comes with Windows Forms (WinForms), a large and growing library of user interface objects which, as objects, can be readily extended if needed.

Downside: The ACCEPT statement goes away in Clarion.NET, the Window structure changes, and the event handling model is a bit different.

Upside: Thanks to reflection, the new Clarion.NET IDE will be able to use any WinForms-compatible .NET control automatically.

Database access

.NET's data access model is largely based on ADO.NET, although there is some very funky stuff going on in the [LINQ project](#), which promises a more generalized query language which you can use on a wide variety of data sources including SQL, XML, and even the objects you use in your own applications.

Here's Bob Z on [data access in Clarion.NET](#):

The application architecture generated by the templates divides your application into layers: Presentation layer, Business logic layer (rules), and Data access layer. The templates allow you to make the decision on what type of data access layer you want to use. You could choose to have your data access layer generated using the familiar Clarion data access model. All of the Clarion data entities (FILE, RECORD, VIEW, QUEUE and GROUP) support data binding to any .Net control that provides for data binding. There are some obvious advantages to this model (and some not so obvious advantages). Or you can choose to have your data access layer generated to utilize the ADO.Net object model. We have DataAdapters for both FILE and VIEW structures that plug right into the ADO.Net object model, so you can happily work with what you know or the new ADO.Net objects. In the future you'll have the option to generate the data access layer utilizing an ORM model like nHibernate or Gentle.Net, or when its released using LINQ.

I'll have more to say on the multi-tier approach later, but for now take note of the mix of data access methods.

Downside: New methods of data access mean new opportunities for bugs. It'll take a while to get everything smoothed out.

Upside: The opportunity to leverage the existing drivers is definitely a plus for migration. And being able to use stuff like [LINQ](#) and [nHibernate](#) is huge, not to mention direct ADO.NET access.

Object orientation

The Windows API is a procedural API; the .NET framework is object-oriented, with no built-in provision for purely procedural code. Yet Clarion is a hybrid language which lets

you create either a purely procedural application or one with a mix of procedures and classes. Does that mean procedural code goes away in .NET? I don't think so, and I haven't heard anything from SoftVelocity to suggest that's the case - quite the contrary. There's no reason that Clarion.NET can't generate object-oriented IL code from procedural Clarion code.

Downside: While procedural code will probably still be quite possible, getting the most out of .NET means using the supplied class library as well as third party class libraries. Clearly there's a learning curve coming for anyone who doesn't yet know object-oriented programming.

Upside: The .NET base class library is organized by namespaces, which makes it a lot easier to find what you're looking for than wading through a big jumble of API calls. And there is way, way, WAY more functionality in the .NET framework than in the Windows API.

Using variables

In .NET everything is an object. Some simple data types are special "lightweight" objects and are treated in code just like simple data types, but under the hood, they're objects. And you can also convert those simple data types to full-fledged objects and back again, as needed. This may happen automatically without you realizing it. As Jim Kane recently pointed out in the newsgroups, this type conversion, called [boxing](#) and [unboxing](#), is an important issue.

Downside: Misunderstanding type conversion can lead to bugs and performance issues.

Upside: Being able to handle simple data types as objects has many benefits. I particularly like working with string objects which contain all of the commonly-needed string functions like clipping, searching, replacing, etc.

Application design conventions

Clarion as we now know it is built on the client-server model. The application is the client, and requests data from the server. Client-server is still a valid model for many applications, particularly those encountered by Clarion developers, but there are other models with greater flexibility and better maintainability and scalability.

In a web-oriented world, applications are more commonly divided into different layers, or tiers. You'll often see what's called a Model 2, or [Model/View/Controller](#) architecture. Above, Z talks about the data access, presentation, and business layers in Clarion.NET applications, which very roughly correspond to MVC. The idea, in either case, is to split out the distinct parts of the application into layers and enable communication between those layers. Clearly the template team has been hard at work:

[The template design](#) accommodates multiple target platforms (win, web, mobile), with a high degree of shared common code, and a great deal of flexibility in the data access layer.

Downside: Just because it's new doesn't mean it's better – client/server still works fine for many applications, and a significant design change to the templates means more bugs to fix.

Upside: Multi-tier applications are a force to be reckoned with, and Clarion needs to start playing with the big boys.

Web development

Clarion/TS/SoftVelocity has a long history of useful, but not-quite-there web development tools. Clarion.NET will be able to leverage Microsoft's Web Forms with ASP.NET, though the details of how that will work haven't been revealed.

Downside: Yet another new technology for Clarion, and one that leverages the above-mentioned multi-tier approach. That's a lot of new stuff to bring together.

Upside: Potential here for the first really serious contender for web development in Clarion.

Clarion language changes

As I mentioned earlier, there doesn't seem to be any reason why Clarion should have to lose its hybrid (procedural/OOP) nature in .NET, but a few features of the language will be going away. These include the ACCEPT statement, the WINDOW and REPORT structure (replaced with WinForms equivalents), and the . (period) shorthand for END.

Downside: Old habits die hard

Upside: The . (period) shorthand looked sloppy to me anyway (although I'm sure hordes of Clarion programmers will disagree). Also going to the WinForms window structure means there's one less wad of custom editor code to maintain.

Mobile device support

The .NET platform is available in a number of flavors, including one called the [Compact Framework](#), which is designed PDAs, mobile phones, set top boxes and the like. Support for mobile devices is something that comes up regularly in the newsgroups, and Bob Z has [blogged](#) that Compact Framework support will be in the first release of Clarion.NET.

Downside: This is more brand new stuff built on other brand new stuff, so there will undoubtedly be some growing pains.

Upside: As with a good web dev tool, this is something a *lot* of Clarion developers can use.

Resources

To learn more about .NET and Clarion.NET, keep your eyes on the following Clarion-related resources:

- [The SoftVelocity blogs](#) – they don't get updated often, but that's the source of the most detailed information to date.
- [The SoftVelocity forums](#) – there is [one forum for Clarion.NET](#) and it's pretty quiet at present as are the web forums in general.
- [The SoftVelocity news groups](#) – all of the Clarion newsgroups (some of which are quite busy with over 100 posts per day) plus a feed of comp.lang.clarion. There is one Clarion.NET newsgroup from the last DevCon but it's unused at present.
- Clarion Magazine, of course! A couple of places to watch:
 - [The ClarionMag blog, C7/.NET topics](#)
 - [The .NET articles index](#)
 - [The .Net interop articles index](#)

- [The Clarion.NET articles index](#)
-

[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 is a member of the American Society of Journalists and Authors ([ASJA](#)).

Reader Comments

[Add a comment](#)

- [» Hi Dave, Nice article, I have just a couple of comments....](#)

Clarion Magazine

Signing Your Applications: New Challenges

by Jane Fleming

Published 2007-01-12

So you get a call from a customer. "When I run your program I get a screen telling me it may be dangerous! It worked fine last week."

"Argh!" you think. "I'm digitally signing my programs now. What's going on?"

"Have you just installed Internet Explorer 7?" you cannily inquire.

And you've figured out your customer's problem.

In this article, I'll describe several possible approaches to dealing with the warning messages: a couple of ways to use a general solution that trusts all computers on your local network, a more granular method where you specify what specific computer(s) you want to trust, and a bit of an 'end run' that's more granular still.

In Microsoft's (commendable) efforts to protect computer users, they've added additional security with various Service Packs and updates. But who would have thought that a new version of IE would affect how you run programs on your computer even when you don't have IE open? However, if your computer is part of a network and you are running a program that resides on another machine's hard drive, IE's security will have its fingers in the pie.

A little while back I wrote a couple of articles about *signing* program files, and how signing is becoming more important as computer security awareness increases. In spite of my accurate and heartfelt disclaimer that I'm far from an expert on this subject, I've since had communication with several folks who've been encountering the new post-IE7 warnings.

I certainly don't have all the answers, but will share a few insights that have come from those

conversations and from some reading and experimenting. I hope that any of you who have developed other solutions will post comments to this article, write your own articles, and communicate on the news groups.

A little background

At this point in the history of computer security we rather expect to be warned when we download a file from the Internet. And, indeed, the warning boxes I showed in the code signing articles reflected that dynamic. If you have IE7 installed, you'll see similar warnings if you run an application from a server on your local network. But unlike the screen shots in my prior articles, these messages don't identify themselves as originating with Internet Explorer.

If the program you're trying to run is not signed, you'll see something like Figure 1.



Figure 1. Unsigned program warning

If the program is signed, you've moved from a red warning icon to a yellow one (Figure 2), but all is not warm and fuzzy. And to add to a user's confusion, the warning refers to the Internet even though the user is trying to run the program from a server on a local network.



Figure 2. Signed program warning

Turning it off manually

David Johnson was the first developer who communicated with me about this issue. The solution he found and showed me involves changing a single setting in Internet Explorer.

To my amazement, I found that (in XP SP2, at least – I'm not yet running Vista) even an ordinary user has permissions to change these security settings. The fact that ordinary users *can* make these changes leads to a caveat – in that the settings we're going to look at are maintained in the Current User portion of the Windows Registry, they will be *different for each user* who signs on to a computer. So if multiple people with individual user accounts use a given computer at different times, *each of them will have to make the changes*.

To make the changes, open the Tools menu in IE7 and select Internet Options (Figure 3).

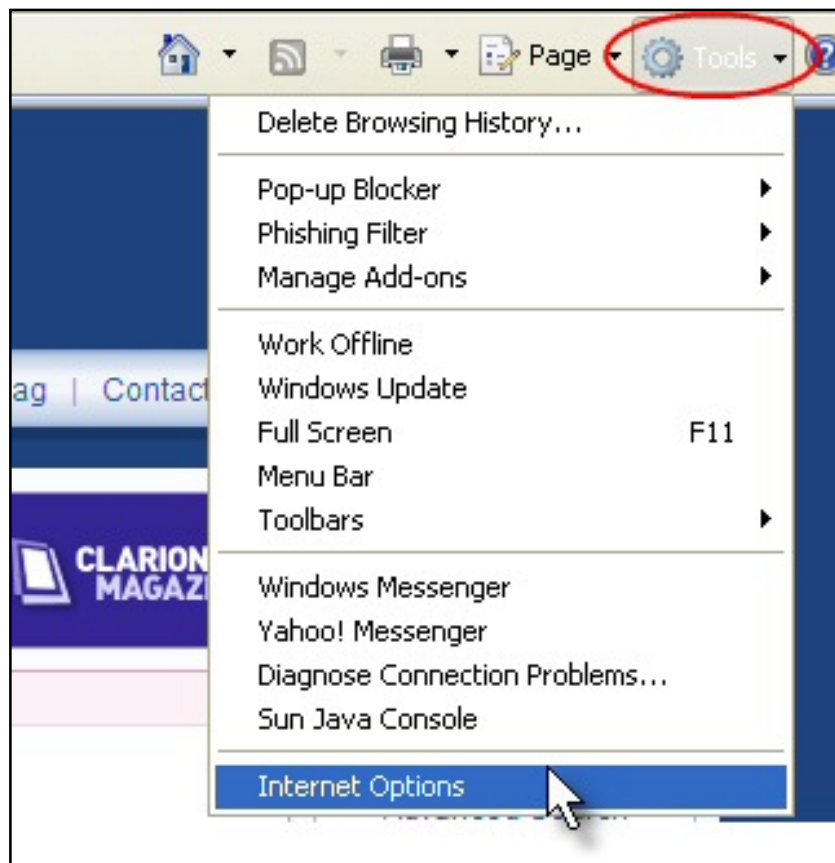


Figure 3. Select Internet Options

Select the Security tab. Click on the Local Intranet icon. Then click the Sites button (Figure 4).

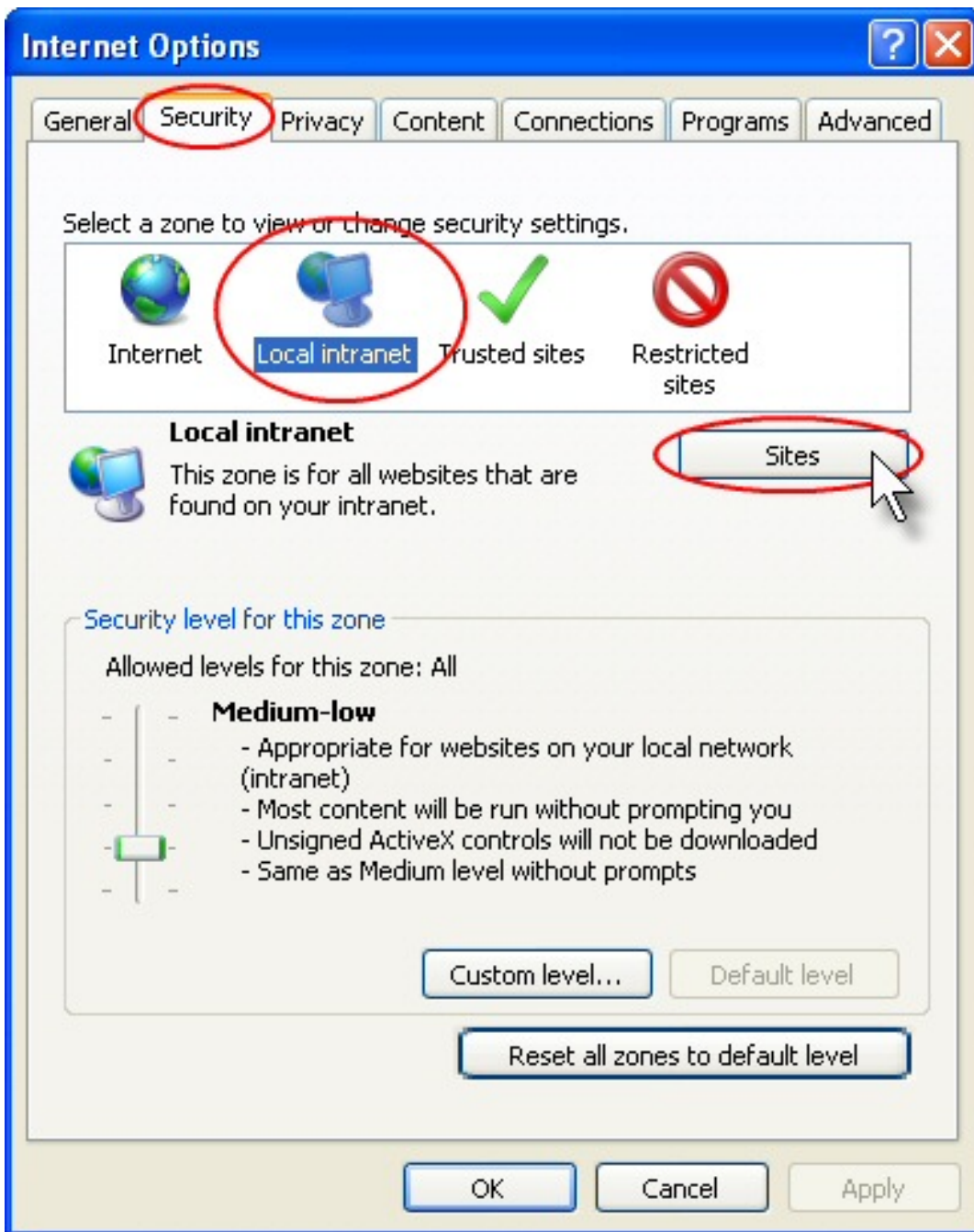


Figure 4. Security options tab

The default setting for a new IE7 installation is shown in Figure 5. Note that the Automatically detect intranet network box is checked and the others are checked but disabled.



Figure 5. Default local intranet setting

The easiest solution is to clear that first check box. Leave the others checked (Figure 6). And keep clicking OK until you've exited the configuration section.

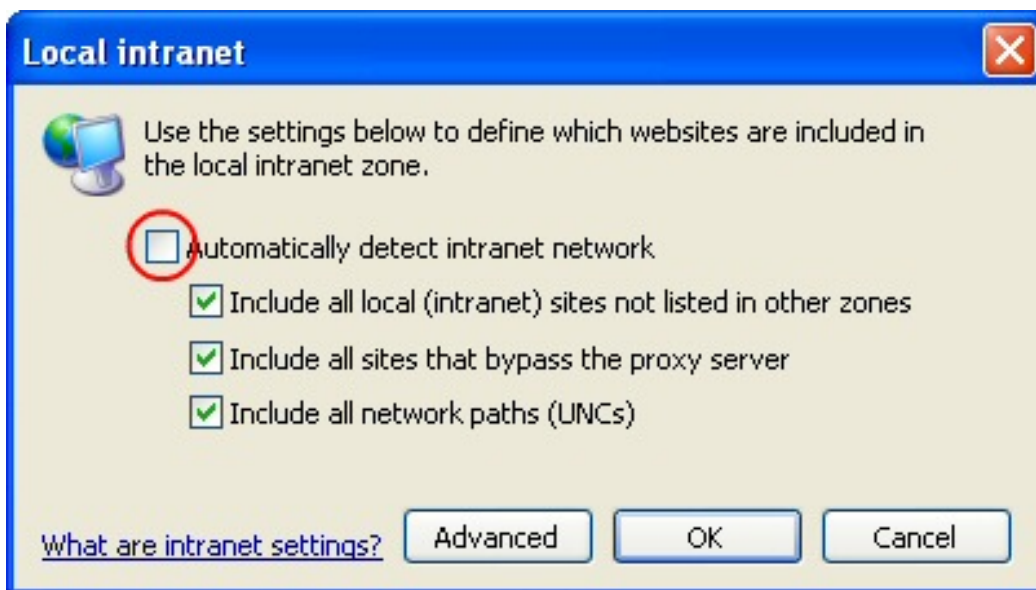


Figure 6. Check box cleared

You should now find that you can run an application from another machine on your local network and not get that warning window each time.

Turning it off with code

Jeff Slarve took the IE7 settings option a couple of steps further. As a first effort, Jeff came up

with a Camtasia-created 15-second instructional video to show his users how to change the setting to what's shown in Figure 6.

Then he spent some time using Regmon to see what was changing in the Registry, and found the flag in question. As mentioned above, the setting is in the Local User portion of the Registry, so it will need to be made by each account that uses a specific computer.

Jeff wrote a little app to detect that setting and offer to change it for the user. His PRJ file and source code are included in the source zip file linked to at the end of this article.

Turning it off granularly

If you have users who don't want to trust all computers on their internal network, you can tell Windows to trust one or more specific servers.

To do so, again click the Sites button (Figure 4), then click the Advanced button. Enter the name of the server you want to trust, in UNC format (Figure 7).

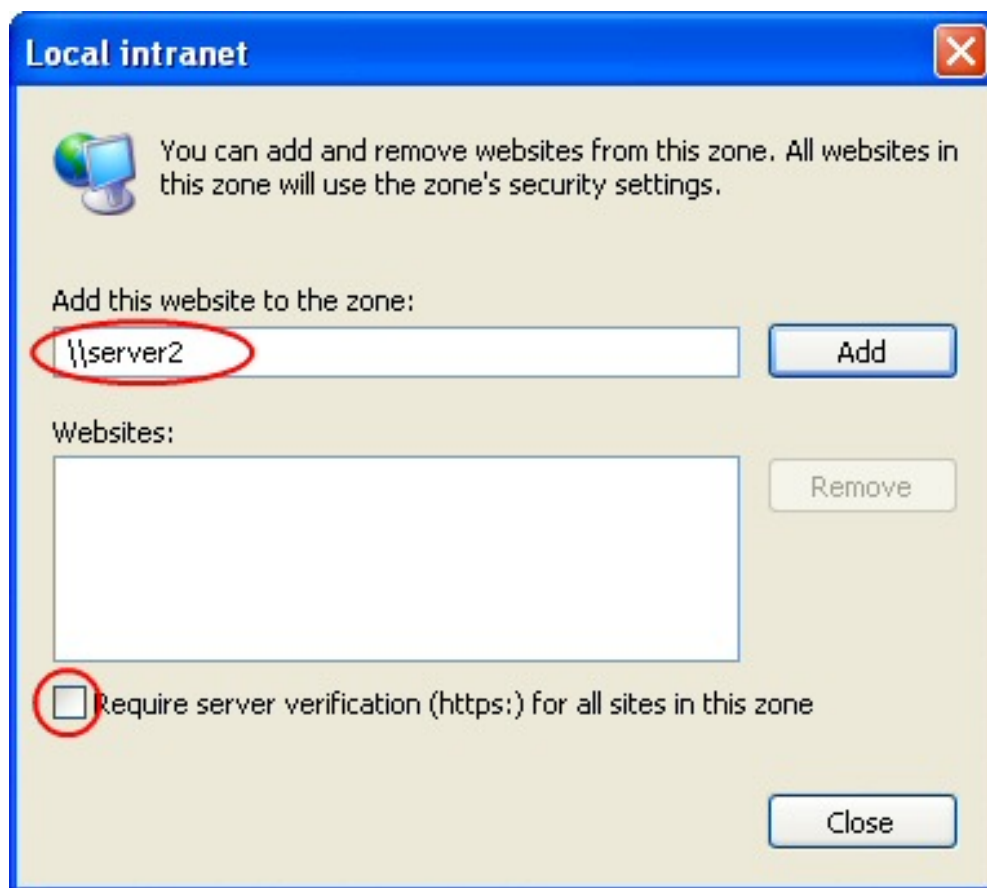


Figure 7. Adding a specific trusted computer

Be sure you clear the https check box.

After you click Add, you'll see the name converted in format and listed in the lower portion of the window (Figure 8).

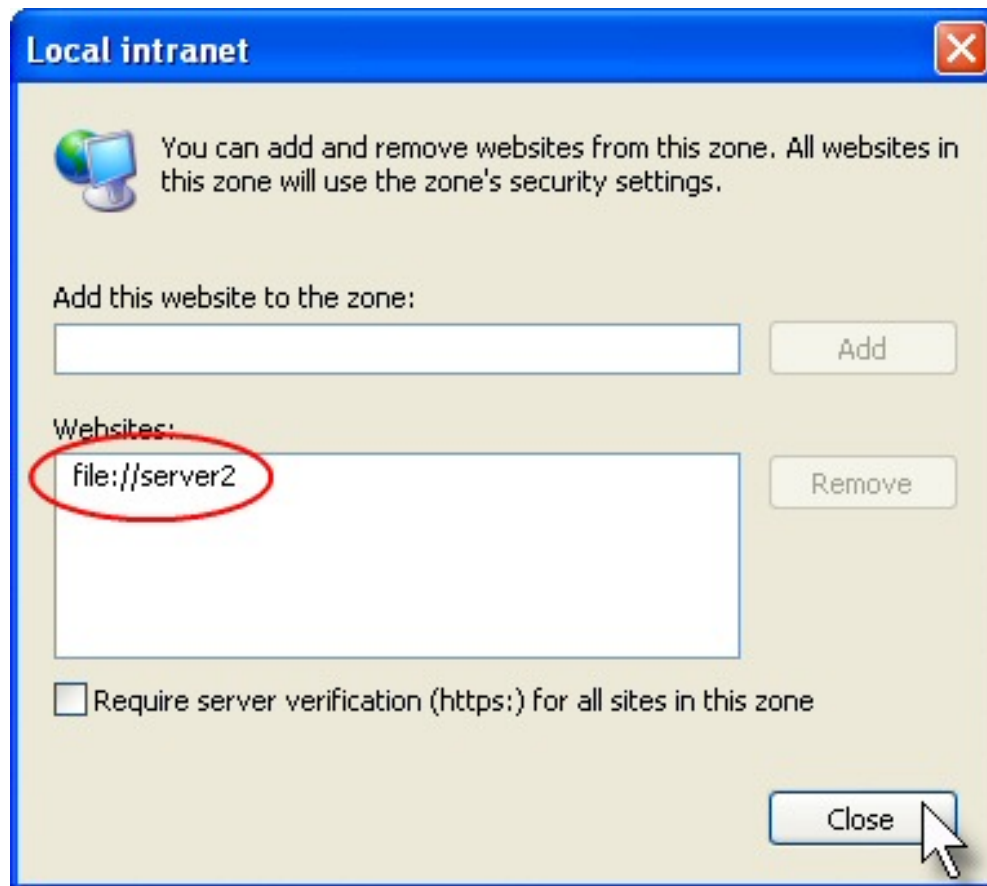


Figure 8. Trusted computer has been added

If you were to look in the Registry, you'd find a new key has been added under `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains` (Figure 9).

The File value of 1 means this domain (actually, Server2 in this case rather than what you'd normally consider a domain) is in the Local Intranet zone.

A wealth of information about these Registry settings is in the "[Rosetta Stone](#)" page I found after I'd explored this part of the Registry enough to have some specific search terms to try on Microsoft's site. If you ever wondered where various parameters get stored if you specify Custom Level for the security level for any of the zones, wonder no more.

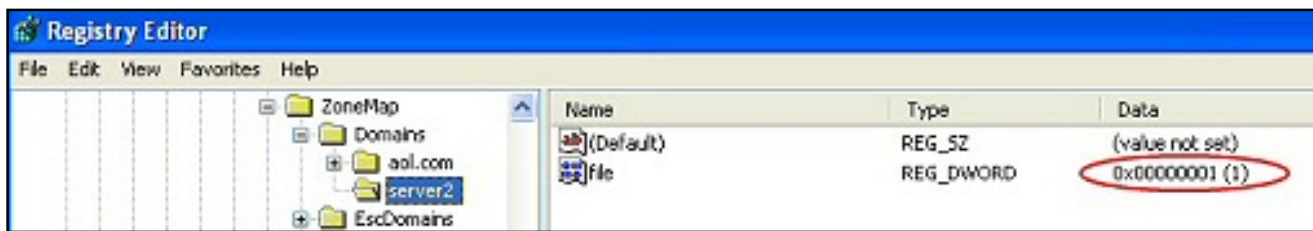


Figure 9. Registry entry added for trusted computer ([view full size image](#))

By the way, if you watch these entries while playing with the interface in IE, remember that Regedit doesn't update in real time. Hit F5 to force it to refresh.

All users?

It seemed it would be convenient if there could be a setting for all users, rather than requiring that each user of a computer go through this configuration process.

I tried adding an identical key into the ZoneMap section of the HKEY_LOCAL_MACHINE section of the Registry, but was not able to get it to work.

I also tried setting the Autodetect value in HKLM to zero, but that didn't work on my test machine either.

An end run - my application only

After I first installed IE7 I wasn't able to reproduce this "are you sure" problem that people had been talking about.

But then, when I log on to a computer the first thing I do is open a couple of command prompt windows.

Then, to run an app from another machine I type:

```
net view \\myserver
net use z: \\myserver\myshare
z:
myprog
```

And everything runs fine without the warning.

So this gave me an idea. How about running an app using a batch file on the client? And... it works!

You can run a network app by batch file one of two ways.

1. Map a drive. This is probably best, as it gives you a handle for the "Start in" section of the shortcut that you'll create in a moment. Assuming you have already mapped a drive (perhaps creating a persistent mapping), just change to that drive and invoke your program. The batch file might consist of these two lines:

```
@z :  
@myprog.exe /anystartupswitch<
```

2. Use a UNC name. If your program can find its data, this is another alternative. You don't need to map a drive. Just create a one-line batch file:

```
@\\myserver\myshare\myprog.exe /anystartupswitch
```

Remember that an @ at the beginning of a command in a batch file prevents that command from "echoing" to the user's screen. If you have more than a line or two, you can put a command at the top of the batch file to turn off echo for the entire batch file:

```
@ echo off
```

So you've made a batch file and tested it from the command line. Now you make a shortcut on the desktop. There's a cosmetic downside - the user sees a DOS window open, and that window stays open the entire time your invoked program is running.

But that's easy to fix in the shortcut you created to the batch file.

The key here is to tell the shortcut to run Minimized (Figure 10). The DOS window won't appear and your app will open with whatever its regular window size setting happens to be.

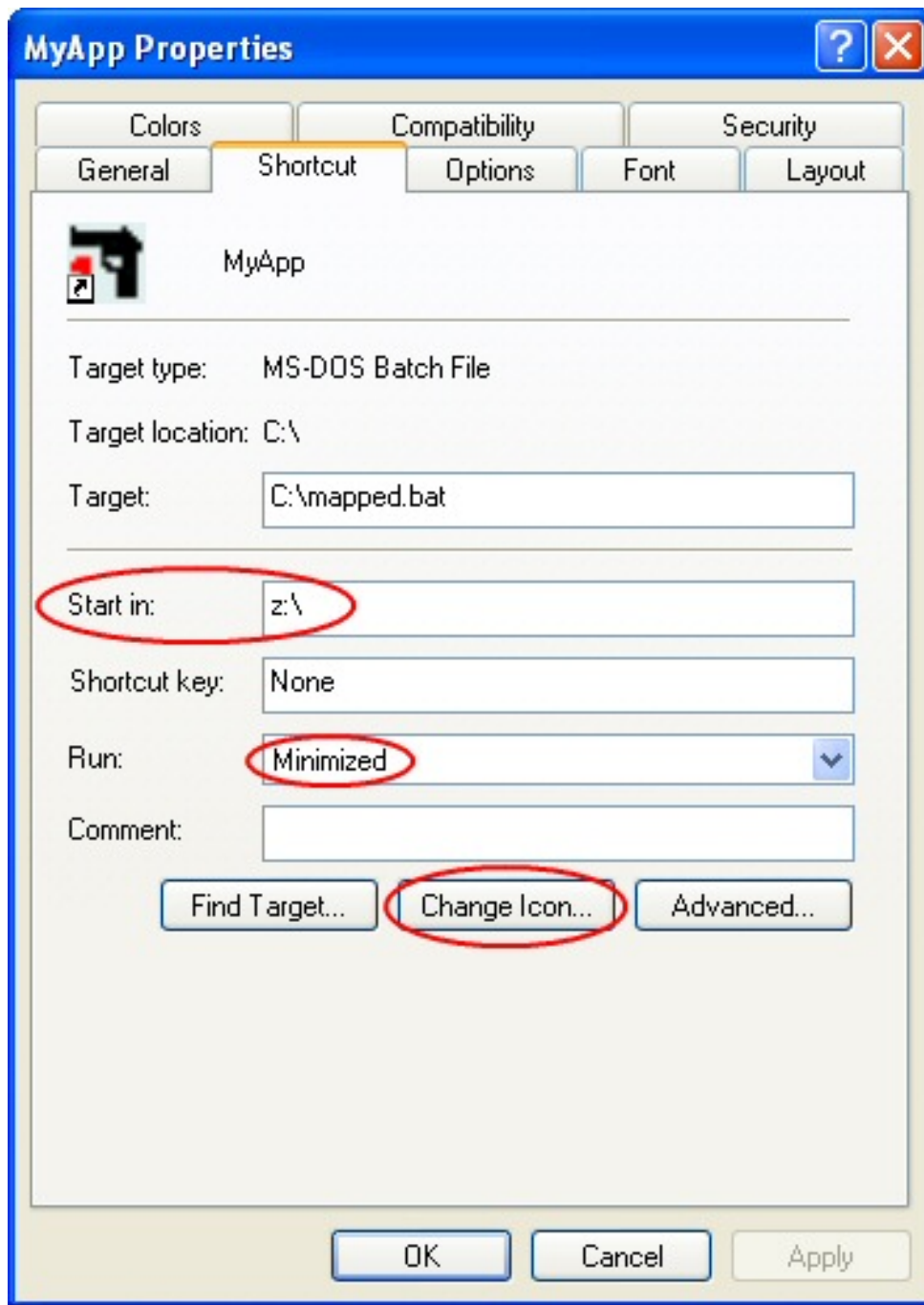


Figure 10. Shortcut for batch file

Because I've mapped a drive for this example, I'm specifying that drive (Z:) in the Start in: field. Depending on how your app finds its data, that may or may not be of value.

To get away from the generic "gearbox" icon Windows uses for shortcuts to batch files, click the Change Icon... button and browse to the icon you want to use. You can select an .ico file on the local computer, or browse to an icon on your mapped drive – whether that is a separate icon file or an icon embedded within the application you're invoking (or even embedded in a different application). An advantage to using the second method for the icon (from the mapped drive) is that if, for some reason, the server is not available and the shortcut can't read

the icon, it will revert to a blank window icon (Figure 11). Instant network diagnostics!



Figure 11. Batch file shortcut when network connection isn't available

For SetupBuilder users

If you know the mapping you're going to use, you can also create the shortcut in SetupBuilder® and specify the icon you want to use.

As for "real" networks...

It occurred to me that there is a fly in the ointment as far as having users change security settings in Internet Explorer. If a computer is a member of an Active Directory domain, it's trivial for a network administrator to use Group Policy to remove the Security tab altogether from IE on all computers on the network. An administrator can also use the [Internet Explorer Administration Kit](#) to deploy custom images of IE.

To test this circumstance I cobbled together a small Active Directory network – one Windows 2000 domain controller and a Windows 2000 member server. Both machines have the latest service packs and updates.

Creating accounts on the servers with the same name and password as one I use on the laptop enabled me to access their shares using my XP Pro SP2 laptop. I left IE 7 at its default settings and got the "are you sure" window each time.

Then I joined the laptop to the domain.

Now when I signed on with a domain user account (regular user, no special privileges)... voilà. No more warnings. With the default IE settings!

Sooo... this is perhaps why Microsoft hasn't been deluged with complaints. There's a certain level of trust invested in computers that are a member of the domain, and the warning doesn't appear.

To verify, I removed the laptop from the domain and the "are you sure" behavior returned when I tried to run a program. Joined it to the domain again and no warnings appeared.

I have not had an opportunity to test this with Server 2003 Active Directory.

I also haven't tested it with any *nix networks. In that those boxes would normally be used only as file servers, I presume the workstation wouldn't have the inherent trust it develops for machines in its own security domain.

David Johnson's problem was using a Snap server for the program, so obviously that doesn't have the security implications of a domain structure either.

I also haven't done any experiments with NDS (Novell) networks.

Miscellaneous security annoyances

You've probably noticed that Microsoft decided not to trust compiled HTML help (.CHM) files a while back. The folks at *Help and Manual* have updated their [free tool](#) to change that behavior. Their new version now supports Vista.

There's been another change to block downloaded files. If you download an .EXE or .CHM file from a web server, when you double-click it to run it from your hard drive you'll first get the "may be dangerous" warning window. If the file is an .EXE and is not signed, you'll see a window like Figure 12. Note that the difference between this and Figure 1 is the "Always ask" check box. If the file is signed, you'll see a screen similar to Figure 2, but again with the "Always ask" box.



Figure 12. Download from Internet, running from local hard drive

If you download a compiled help (.CHM) file onto your hard drive and double-click to run it you'll see Figure 12 (a .CHM file cannot be signed). If you *do not* clear that check box, the help file's contents will be masked (Figure 13).

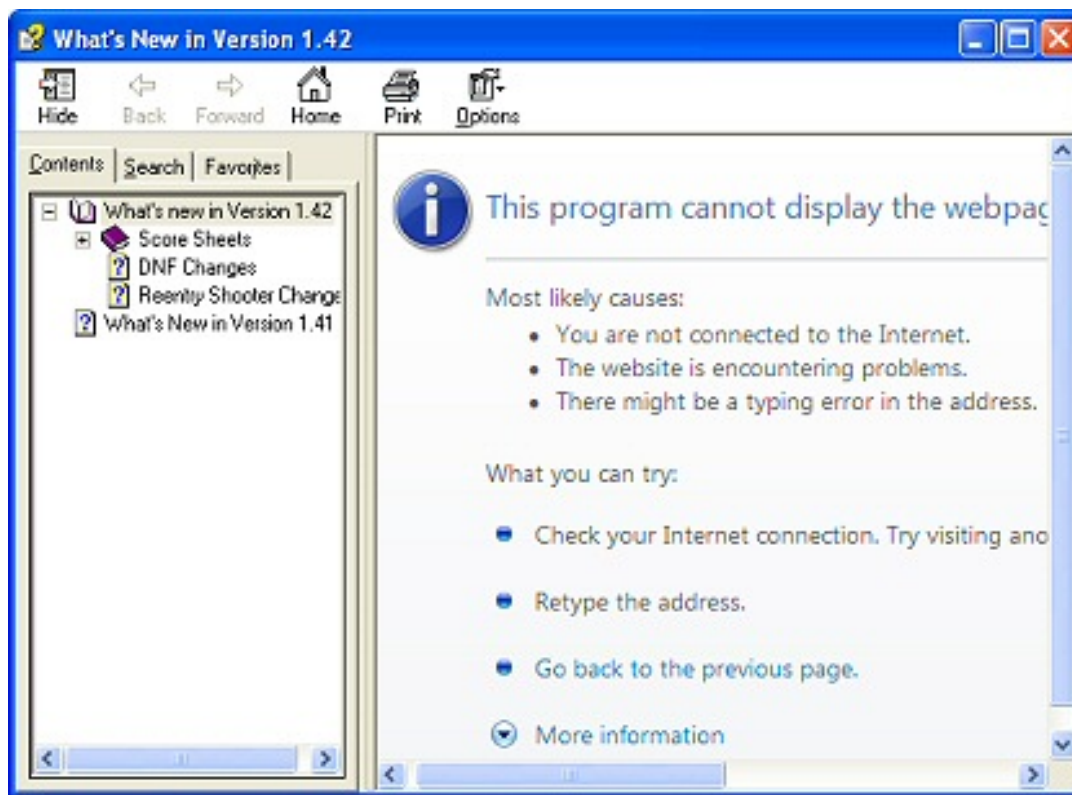
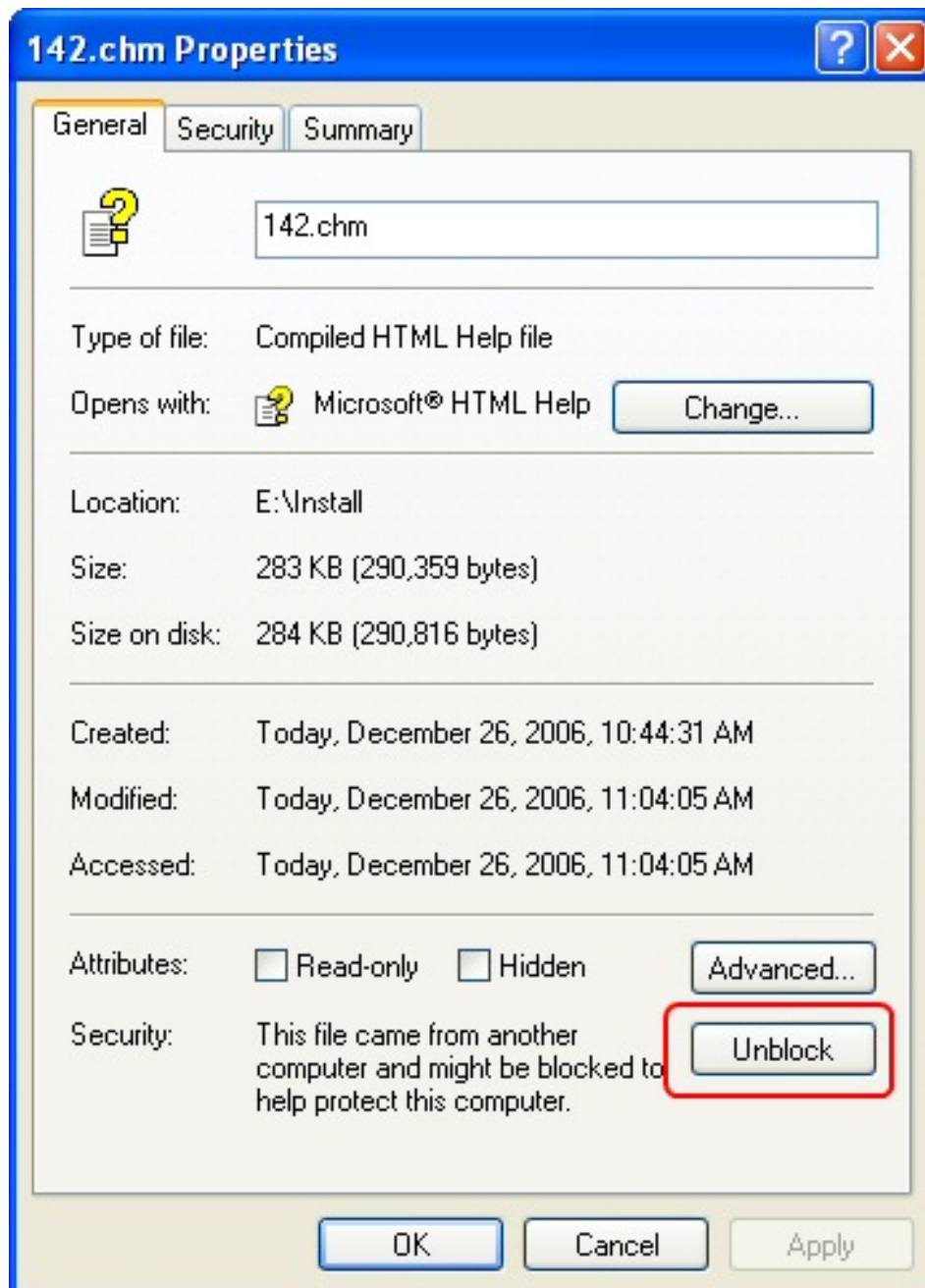


Figure 13. Compiled help file with content masked ([view full size image](#))

Again, the "helpful" suggestions will likely confuse a user, because he knows he's not trying to open the help file from the Internet.

If you have *not* cleared that check box and look at the file's Properties page, you'll see a new button – Unblock. Clicking that button will make the file's contents available from that time onwards, and it will no longer bring up the "may be dangerous" window. If you (or your user) clear the "Always ask" check box on the warning screen, that has the same effect as clicking the Unblock button.

**Figure 14. The Unblock button**

Conclusion

As I used to tell my networking class students, "Be glad computers aren't easy or a lot of us would be out of jobs."

I've tried to describe a few ways of running your application from a network server without being bombarded with warnings every time. As mentioned in the introduction, I don't claim to have definitive answers and hope that you'll pass along whatever you may learn.

Additional Reading

- [The "Rosetta Stone" of IE security Registry entries](#)
- ["How to use security zones in Internet Explorer"](#)
- [Enhanced browsing security in XP SP2](#)

[Download the source](#)

[Jane Fleming](#) is a college dropout who subsequently lived four years in Europe, a year and a half in Mexico, and three years in India, and later taught yoga for a living in California (she's been vegetarian since 1970). She developed circuits and wrote assembly code for several embedded microcontroller projects during the 1980s. She began using Clarion Professional Developer for in-house projects back when Clarion was running display ads in InfoWorld and has used it very intermittently since. She is a former Microsoft Certified Trainer and taught Microsoft and Novell network administration at a business college for four years. Now widowed ten years, Jane plays [classical piano](#) and has found her métier as a semi-retired NRA-certified pistol instructor.



Reader Comments

[Add a comment](#)

- [» I can't find "...the source zip file linked to at the end...](#)
- [» My apologies - I've updated the article with the link. You...](#)
- [» Hi Jane Sigh - MS again. Now how long before they close...](#)
- [» Apps delivered with a Novell Application Launcher will not...](#)
- [» Apps delivered with a Novell Application Launcher will not...](#)

Clarion Magazine

Review: Clarion Training on Demand - SQL Programming Series

by Russell Eggen

Published 2007-01-10

SoftVelocity recently released Clarion Training on Demand (CTOD) [Course VII – SQL Programming Series](#). This installment in SoftVelocity’s training series begins with a refresher on SQL and progresses to advanced topics, covering a wide variety of backend databases as well as ODBC.

In a departure from SoftVelocity’s normal shipping methods, this course is available as an electronic delivery instead of a CD product.

Purchase and installation

When you complete your purchase you’ll receive an email with download instructions. You either download a single file, which is 80.5MB in size, or a series of smaller files, about 9MB in size each. I chose the single file option; my download speed in Tampa FL was close to 400KB/s.

Installation was very straight forward. The install program asked for a serial number and then continued uninterrupted to completion.

To start any lesson in SV’s training series you run the Clarion Training on Demand Launcher. This application lists all the titles published, but a gold star appears next to lessons actually installed on your computer. The drop down control allows you to filter based on the course you want. The Figure 1 shows the SQL Programming Series lessons,

along with a description of the currently selected lesson.



Figure 1. The Clarion Training on Demand Launcher

You study the lessons in sequence or jump to a lesson that is important to you. If you are new to SQL or wish to do a review, then I recommend starting at the beginning.

Although the available lessons are shown in the CTOD launcher, the lessons themselves run in your browser. I use Internet Explorer 7. Depending on your security settings, you may get a warning about running active content. If you allow active content, then you will see controls at the top of the page. The launcher's help file has more information regarding allowing active content."

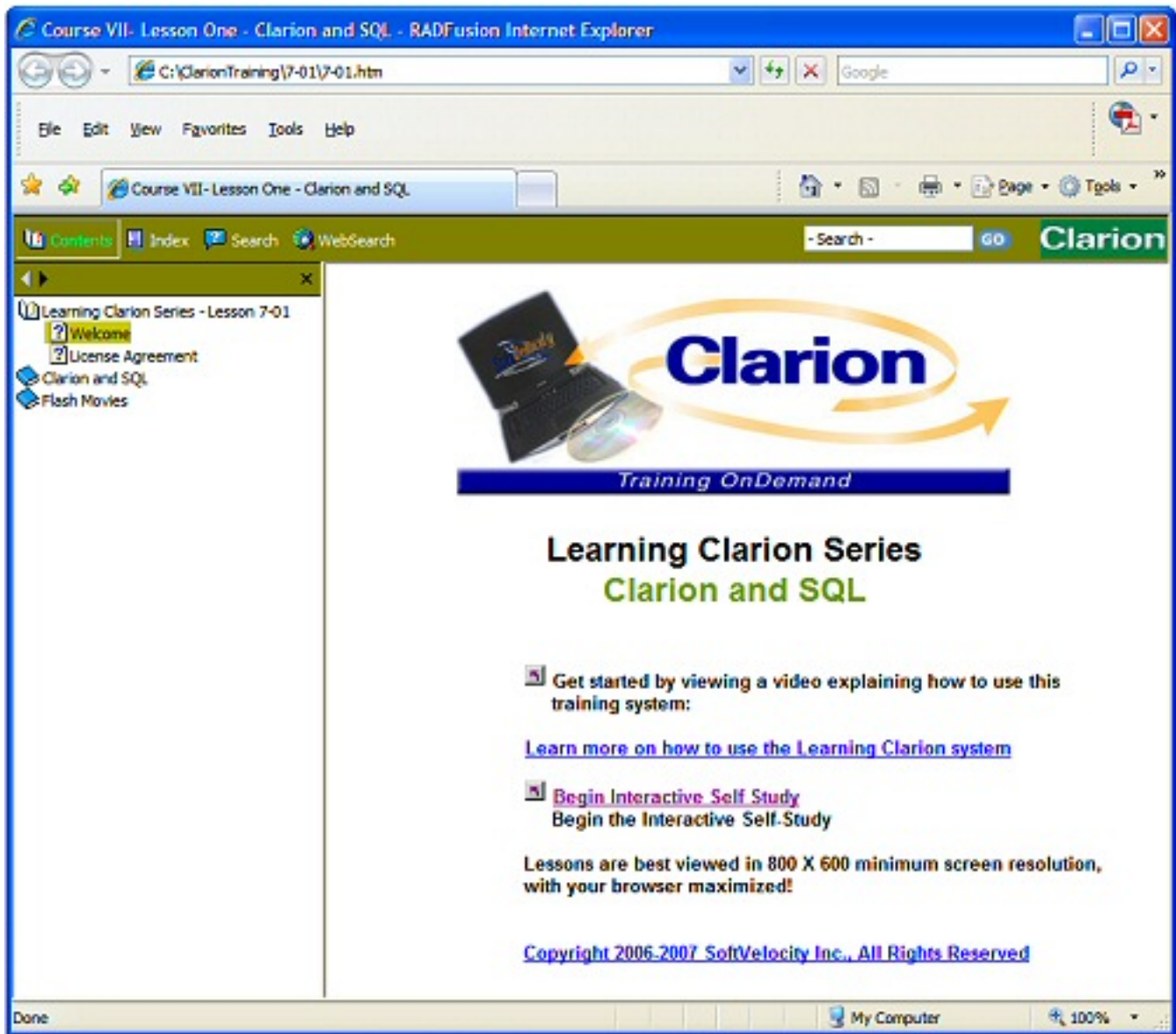


Figure 2. The opening lesson ([view full size image](#))

As Figure 2 indicates, 800x600 is the minimum recommended resolution. The page looks fine at higher resolutions.

Each lesson is a mixture of text and video, much of it narrated. The lessons also work well as reference material, and every movie is available quickly on the navigation bar.

In the past, SQL lessons concentrated on one database, usually MS SQL or ODBC. This course is more extensive, with specific coverage of MS SQL, Pervasive, SQL Anywhere (Sybase) and Oracle, for which Clarion has native drivers. The course also covers using ODBC to get access to MySQL, PostgreSQL, and Firebird databases. Links to these readily available databases are included.

Bob Foreman is the narrator; his voice is easy to listen to and the recording quality is top notch.

There is a dedicated newsgroup (the Clarion Classroom) available to you in case you need to ask any course related question(s). Course corrections and minor updates are also posted there.

What is SQL?

The first lesson begins by answering the question "What is SQL?" Next it moves on to the benefits of SQL and also lists a few general guidelines as to what ask yourself or your clients about future needs; the answers to these questions will help you determine if an SQL version of your application is in your future.

The lesson quickly moves into basic SQL commands and shows some simple examples and, more importantly, why you would use these commands. This section covers the basics and does not go deep into the platform-specific flavors of any one command. (Editor's note: As Clarion developer Mike Gorman, Secretary of the ANSI Database Languages Committee, points out, there is no such thing as *standard SQL*.)

For example, the course shows a simple use of SQL as in `SELECT CustomerName FROM Customer`. Anyone who has any experience with SQL knows the `SELECT` statement can and does get very complex. This section sticks to the basics, especially at the introductory level.

On the other side, staying simple for too long can cause a subject to become boring. This course avoids boredom by letting you study at your own speed. You can go through the materials as fast or as slow as you want. The subject matter does build on prior lessons so each time you move forward, you get a little more information.

Specialized words are clearly defined *early* in the lesson. For example, stored procedures:

A **stored procedure** is a program (or procedure) which is physically stored within a database.

The lesson then moves on with what stored procedures are, how and why to use them, benefits, etc. After a few more topics there is a tutorial. Again, this is a nice touch as no one can learn on theory alone. The example commands can be practiced on just about any

SQL table you like. All the SQL databases I've ever used come with example databases, which are ideal for study.

The course content is not all SQL statements and commands. Clarion language statements are included as well. By that I mean driver strings, actual Clarion statements that deal with SQL (like SEND) and even property expressions. This helps put SQL statements into a familiar context.

At the end of each lesson is a self test. I highly recommend taking the test. It is multiple choice and if you get the right answer you are greeted by a balloon window saying so. If you get the wrong answer you are likewise informed, but you have a link to go back to the topic so you may review. You won't lose your place if you do since the left navigation bar is visible by default. The self-test is always the last item listed on the navigation bar.

I found a small cosmetic bug in the first self-test. I reported the problem in the newsgroup and within twenty minutes had a fix from Bob Foreman.

As I went through these lessons I was reminded of features and tools that I have let drop out of use, such as the dictionary synchronizer and its powerful data conversion feature (single or multiple tables). This would be a nice tool to use if, say, you have say an accounting package and you want to convert your customer's existing data to an SQL database. The synchronizer can generate a conversion program that converts all of your customer's data to your SQL backend of choice in one go.

I was also impressed with Query by Example and how effective it is with little or no alteration. Movies and example applications illustrate the use of these features.

One particularly nice lesson covers the SQL Advanced tab as seen in Figure 3.

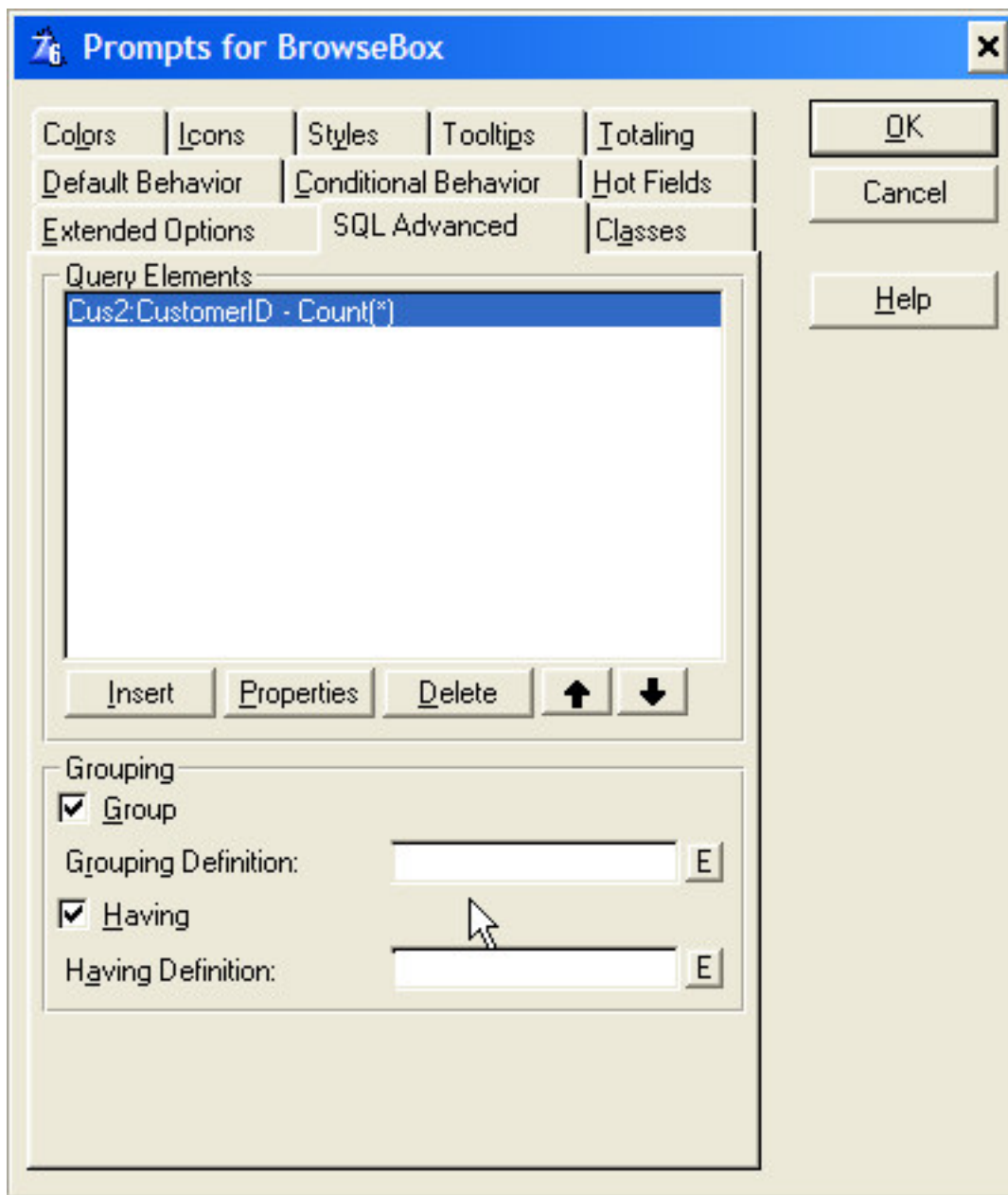


Figure 3. The SQL Advanced Tab

If you use SQL, have you seen this tab before? I have, but to be honest, I've never used it. I've not really bothered to explore what it does, which is a shame. Lesson two has in-depth coverage of the SQL Advanced tab, which lets you add SQL expressions (such as aggregate functions) to the generated SELECT statement.

Dedicated chapters

Each flavor of SQL has its own chapter and rather than go into all the details I'll focus on the MS SQL chapter.

There is one particular section in the MS SQL chapter which will be of interest to anyone waiting for new products from SoftVelocity. This lesson shows screen shots of Microsoft's *Server Management Studio* (SMS). The first paragraph after this illustration is this:

SMS requires the 2.0 .NET Framework. Actually, much of the window navigation and style of SMS looks very similar to Clarion 7 and the Clarion.NET products, so you should quickly feel comfortable with the new interface.

The MS SQL chapter briefly mentions server setup issues, particularly regarding firewalls, covers some more synchronizer features, and shows how to enable a login window by setting a driver option. There is a nice list of all the MS SQL data types and their Clarion equivalents. One data type that I did not expect was XML, which Clarion handles as a STRING.

SQL Server 2005 has a new schema for table names. Prior versions of MSSQL used a database.table naming convention, e.g. `dbo.Customer` and `dbo.Employee`. The new schema allows you to group your tables using a convention like `Person.Customer` and `HumanResources.Employee`. You can see examples of this in the demo database. The Synchronizer supports this new schema (as of 9055).

A discussion of keys is next and has some useful tips on how to get around problematic naming conventions (in case you have to work with a Database Administrator or otherwise are not allowed to change the table attributes).

The next MS SQL section contains a good discussion on stored procedures and when and why to use them, complete with an example application and a link to the example database on Microsoft's site.

No SQL course would be complete without a discussion of triggers. The section on MS SQL triggers contains a flash movie showing how the listed trigger was created, and includes another link to Microsoft's site for more information.

The TURBOSQL switch is next; this is a potentially dangerous feature (like string slicing) and its use demands some care. The lesson offers usage advice.

Looking briefly at the other chapters I saw the same basic format and information, but the driver-specific chapters are not exact duplicates. Each chapter contains information

specific to that particular driver. For instance, the chapter covering the ODBC drivers discusses DSN-less connections, a subject that doesn't apply to the Accelerator drivers.

SQL is a language, and debugging can be an issue for SQL statements as it is for Clarion programs. There is a nice discussion on how to use driver tracing.

There are lessons on using the IMDD and DFD drivers with SQL, complete with example applications, as well as discussion of open source databases with specifics and links to various sites.

The last lesson in this section targets the ODBC API. As with any well designed course, there is an example application. This one tests if a connection to a database is present. An explanation of the code in this example is, of course, included.

The content of this course is quite exhaustive yet very easy to get around and navigate. I would suspect that a Clarion developer could do an intensive study of this course in perhaps three days of full time study.


My only wish is that there was a PDF included with this course as it would be useful for a future reference guide. Not having one does not really take away anything as the course is well researched, comprehensive, and does not skimp on details. I would not mind seeing more flash movies as well, but the ones provided are quite detailed.

Summary

SoftVelocity's SQL Course is a well planned product that flows easily from one topic to the next, building on what was previously discussed.

This course is on sale for \$199 until January 15, 2007 and you can order direct from SV Sales at 1-800-270-4562, or from [SoftVelocity's web site](#). The content is appropriate for beginners as well as seasoned pros, and is a nice reference for anyone developing SQL applications in Clarion.

Bob Foreman informs me that the course will be updated as newer versions of Clarion are released; any needed changes will be made available to students. I recommend this product enthusiastically!

Rating: 

[Russ Eggen](#) has been using Clarion since 1986. Until about 1996, he was using it for business applications, mostly accounting programs. Afterwards he joined TopSpeed as a consultant, and later as an instructor. He was a founding member of SoftVelocity when that company formed from TopSpeed in May 2000. He left SoftVelocity in January 2001 and now works for his own company, [RadFusion Inc.](#) He still teaches and lectures, and is currently working on a new book and setting up a local Clarion classroom. Russ enjoys flying, scuba, and applied philosophy, and with great effort you might coax him into political discussions.

Reader Comments

[Add a comment](#)

- [» Hi Russ, Good review! I think these courses can be...](#)
- [» Viggo, Thanks! For others in a similar situation, here...](#)
- [» Russ, I certainly agree that this course is well worthwhile...](#)

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](#)

Let's see your pics!

[Direct link](#)

Posted Tuesday, January 16, 2007 by Dave Harms

Photographer to the (Clarion) stars Leroy Schulz is again running the Andrew's Kitchen photo showcase. Andrew's Kitchen, if you don't already know, is one of several private

newsgroups available to Clarion Magazine subscribers (check your subscription email or [contact me](#) for access). You don't, however, have to be a subscriber to participate in the photo event.

Here's the blurb:

Back for a second year, The Best of Andrew's Kitchen 2006 is meant as a showcase of a world inhabited by Clarion programmers. Join other friendly Clarion developers in showing off what has been beautiful, unique, humorous, touching, memorable, or just plain meaningful in your 2006.

Details:

1. Email photos to mail@frostbytes.ca. *One photo per email please!*
2. Please submit images as high-resolution JPG. (Images will be down-sampled as necessary.)
3. Photos must have been shot from January 1, 2006 to December 31, 2006.
4. Include the following information in the body of your email:
 - A. Title of image.
 - B. Photographer name.
 - C. Location.
 - D. Date.
 - E. Short (2-3 sentence) description.
 - F. Technical data. (Optional.)
5. Five entries per person maximum.
6. Entries must be received by January 31, 2007.
7. Copyright remains with the individual photographer, but will be made visible on the website and/or a downloadable slide show.

View current entries at <http://www.frostbytes.ca/gallery/bestofak2006>.

View last year's entries at <http://www.frostbytes.ca/gallery/ak2005>.

.NET book suggestions

[Direct link](#)

Posted Tuesday, January 16, 2007 by Dave Harms

There was an interesting thread in the newsgroups a few days ago prompted by Arnor asking for .NET book suggestions. Titles put forward include:

- Get Visual C# 2005 Step by Step by John Sharp. ISBN-13:978-0-07356-2129-9 - Russ
- VB.Net remoting handbook by Curran, Olsen, and Pinnock
- Programming Microsoft ADO.NET 2.0 Core Reference
- Programming Microsoft ADO.NET 2.0 Applications, Advanced Topics
- Programming Microsoft ADO.NET 2.0 Step by Step
- Data Binding with Windows Forms 2.0 Programming Smart Client Data Application with .NET
- Beginning ASP.NET 1.1 with Visual C# .NET 2003, C Ullman, J Kauffman, C Hart, D Sussman and D Maharry, Wrox, 2004.
- ASP.NET In A Nutshell, GA Duthie and M MacDonald, O' Reilly, 2003
- .NET Framework Essentials, T Thai and HQ Lam, O' Reilly, 2002.
- Programming .NET Security, A Freeman and A Jones, O' Reilly, 2003
- .NET Common Language Runtime Unleashed, K Burton, SAMS, 2002
- "Expert C# Business Objects" by Rockford Lhotka
- Programming VB .NET A Guide For Experienced Programmers - APress ([free](#))
- Richter's The Applied Microsoft .NET Framework Programming in C#
- ASP.Net Website Programming: Problem - Design - Solution
- Windows Forms 2.0 Programming by Chris Sells
- Windows Forms in Action, by Erik Brown

Dennis Evans also pointed out O'Reilly's [Safari](#), a service that lets you read books online for \$20 per month, while Raymond Maillard suggested the data tutorial course material at [ASP.NET](#). Jim Kane pointed out the importance of understanding [boxing](#) and [unboxing](#).

Ben Brady suggested [Dissecting a C# Application: Inside SharpDevelop](#). This free download is an inside look at the architecture of SharpDevelop, which as you probably know forms the foundation of the new Clarion IDE. Based on the architecture as discussed here there may be some interesting opportunities for Clarion third party vendors to add value to the new IDE.

And yes, I do have a number of articles in the pipeline to help Clarion developers get up to speed on .NET.

Clearing the blogpipe

[Direct link](#)

Posted Friday, January 05, 2007 by Dave Harms

I've had a bunch of small items plugging up the blogpipe since well before Christmas. Time to hook up the ol' Binford 9600 HighVelocity Air Compressor and clear this puppy...

FAWOOOOOMMMPPP!!

[VOX for Skype](#) is an application that lets you use [Skype](#) from your mobile phone. You call your computer via [SkypeIn](#), VOX recognizes you by your phone number and gives you access to your contacts list via a voice menu. You can make calls, retrieve messages, and check for missed calls. Looks like a great way to make long distance calls from your cell and pay only local charges (plus any [SkypeOut](#) fees if you're calling folks not on Skype).

Somebody pointed me to [regender.com](#), a site that flips the gender of names on a specified web page. For your entertainment, get in the Wayback Machine and read [Andrea Guidroz's report on ETC IV](#).

How about a [couple of graphs](#) showing PostgreSQL scaling better than MySQL...

Arnór Baldvinsson's Vordur Install Manager "helps you to manage all your install file information such as serial numbers, passwords, keys, file location

and download information." What other secret products does Arnór have?

Ever hear from users who say your site looks crappy under their browser? Okay, maybe your site looks crappy under all browsers. But to see if you have any browser-specific issues, go to [BrowserShots](#). This nifty web site will create screen shots of the page you specify so you can look for problems without having to install each and every browser. After you submit the job bookmark the resulting page, and check back later for updates (it can take a while for the site to render all the requested versions of a page).

Recent screenshots for <http://www.clarionmag.com/>

This page shows the newest screenshots for a single web address. You can bookmark this page to check for new screenshots later.



Firefox 2.0 on Linux,
2 hours ago



Safari 2.0 on Mac OS,
3 hours ago



MSIE 6.0 on Windows,
3 hours ago

Unexpectedly, [ILoveJackDaniel's](#) is a great place to find cheat sheets for

HTML, CSS, Ruby on Rails, regular expressions, and more. A tip of the hat to luthier Joe Federer (okay, he codes too).

Former Clarion compiler author David Bayliss is a regular reviewer on Amazon. Read his profile [here](#).

Vista's new security requirements have been the subject of much discussion lately in the newsgroups, and Friedrich Linder is one of the reigning experts, having dealt with these issues in [SetupBuilder 6](#). For a little background read this article, suggested to me by Todd Carlson, on [UAC elevation](#). Also from Todd, Paul Robichaux's [Administering the Windows NT Registry](#).

Mike Pickus likes [StExBar](#), billed as "The ultimate tool for Windows Explorer" or "what Microsoft forgot to implement in the explorer". There are some nice customization examples on the [home page](#).

Jeff Slarve points out that in C6 local views and files have thread scope. Before C6 local files and views had procedure scope. This may bite you if you declare local files and views and rely on an implicit close when they lose scope.

I also wanted to point out Bob Z's mention of partial class support in the [update to the roadmap update](#). Partial classes are classes split across several source files, so you can have some methods in one file and some in another. This is a useful trick for code generation systems, although I haven't heard whether the first release of Clarion.NET will employ partial classes this way.