Clarion Magazine

Clarion News

- » SetupBuilder 6.9 Build 2482 Developer Edition Hotfix
- » Base64 Example
- » SetupBuilder 6.9 Build 2481
- » Noyantis ShortcutBar 1.15
- » J-HTML Improved
- » Noyantis PropertyGrid 1.03
- » Noyantis CalendarPro 1.14
- » Noyantis ShortcutBar 1.15
- » Noyantis CalendarPro 1.14
- » FinalStep 2.20
- » CPCS Addons For C7
- » Clarion 6 UAC Fix 1.20
- » CPCS v7.00 Beta
- » Medical Icon Collection
- » SetupBuilder Totorial
- » Mallorca Lite Icons For $10
- » Noyantis PropertyGrid 1.01
- » Beta Test Web Site In Beta
- » Product Scope 7.5
- » PDF-XChange Version 4
- » CHT C7 Video Version 2
- » Noyantis ShortcutBar 1.14
- » Clarion Third Party Profile Exchange January 2 2009 Release
- » Seven Icon Collections For $199
- » DMC Updated Installers
- » Icetips 33% Discount Ending
- » Icetips C7 Installs
- » Icetips Acquires PowerOffice Tools
- » Noyantis ShortcutBar 1.13
- » Oldaer
- » Free VuDeploy
- » Christmas Podcast
- » Lodestar Software Subscription Plan Costs To Increase
- » Clarion Accessory Deployment Demo
- » SetupBuilder 6.9 Build 2454 Pre-release

[More news]

- » Clarion.NET FAQ
- » Clarion# Language Comparison
- » Proper Case in Clarion#

[More Clarion & .NET]

[More Clarion 101]

Latest Free Content

Latest Subscriber Content

Clarion#, C7 Beta Builds Released

SoftVelocity has released a new Clarion# build to beta testers. This release adds support for LINQ, extension methods, and implicitly typed variables. Improvements have been made to the ASP.NET designer and there is a new CSS editor. A number of bugs are listed as fixed. And there's a new release of the C7 beta as well (build 4790). Look for more detailed coverage in Clarion Magazine in February.

Posted Friday, January 30, 2009

Buying A Code-Signing Certificate: The 2008 Version

Back in 2006 Jane Fleming wrote a series of articles about the purpose and mechanics of code-signing and how to negotiate the certificate purchase through the Comodo website (Comodo having been much less expensive than Verisign). Two years later Jane's certificate was in need of renewal...

Posted Friday, January 30, 2009

Managing Global Stuff With The Datafier

In this age of multi-threaded applications, global data is a tricky proposition. Jeff Slarve started looking for a way to manage global data. He started out with a class that managed a queue of variables, and ended up with the Datafier, a class that manages variables *and* expressions.

Posted Thursday, January 29, 2009

Using Vista's Complete PC Backup

Vista Ultimate (like Business and Enterprise) has a feature called "Complete PC Backup" which uses drive imaging technology. Jane Fleming shows how to use this feature to backup and restore Vista drives.

Posted Thursday, January 29, 2009

Managing And Debugging Builds With External Project Files

You may already know that you can add an external PRJ or PR project file to an APP file's project settings. As Rick Martin explains, this capability makes it easy to switch between a release build and a build that includes ASSERT statements to assist in debugging.

Posted Friday, January 23, 2009

Using Multiple Clarion Versions In C7

Years ago it was announced that Clarion 7 would be able to build applications in any version of Clarion. The .APP file would not be backwardly compatible, of course, but the resulting DLLs and EXEs would be built using the templates and binaries of the selected version. Steve Parker explores the multi-version capabilities as found in the most recent CSP beta releases of C7.

Posted Friday, January 23, 2009

Proper Case in Clarion#

Last month Mike Hanson wrote about a class he's long used to implement proper case in Clarion applications. This time around Mike ports MHProperClass to Clarion#.

Posted Friday, January 16, 2009

C7 AppGen CSP Beta Report

SoftVelocity has now released two C7 AppGen betas to the CSP particpants. Dave Harms reports on the status of the beta and the thousand-plus beta newsgroups messages posted since Christmas.

Posted Monday, January 12, 2009

Source Code Library 2008.12.31 Available

The Clarion Magazine Source Code Library has been updated to include the latest source. Source code subscribers can download the December 2008 update from the My ClarionMag page. If you're on Vista please run Lindersoft's Clarion detection patch first.

Posted Thursday, January 08, 2009

[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.

- » Clarion#, C7 Beta Builds Released
- » Source Code Library 2008.12.31 Available

[More free articles]

Clarion Sites

Clarion Blogs

More info • Subscribe now

Printed Books & E-Books

E-Books

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

Printed Books

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

- » Clarion Tips & Techniques Volume 4 - ISBN 978-0-9784034-09
- » Clarion Tips & Techniques Volume 3 - ISBN: 0-9689553-9-8
- » Clarion 6 Tips & Techniques Volume 1 - ISBN: 0-9689553-8-X
- » Clarion 5.x Tips and Techniques, Volume 1 - ISBN: 0-9689553-5-5
- » Clarion 5.x Tips and Techniques, Volume 2 - ISBN: 0-9689553-6-3
- » Clarion Databases & SQL - ISBN: 0-9689553-3-9

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

From The Publisher

About Clarion Magazine

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

Subscriptions

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

Satisfaction Guaranteed

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

Dave Harms

ISSN

Clarion Magazine's ISSN

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

About ISSN

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

Clarion Magazine

Clarion News

Search the news archive

Chicago CUG Meeting

The next meeting of the Chicago CUG will be held Wednesday February 11th, 2009 at 6:30 pm in suite 270, in the Nebo Systems offices. The address is 1 South 450 Summit Ave. Suite 270,Oakbrook Terrace, IL 60181. The sign at the front of the office complex says Summit Oaks. Steve Parker will be giving the presentation on Clarion 7, showingthe latest build with the AppGen.

Posted Tuesday, February 03, 2009

Keystone 3rd Party Tools For C7

Keystone's 3rd party tools for Clarion Release 7.0 are now available. Registered owners can download from the web site. You will need to contact support to confirm your email address and obtain new installation keys.

Posted Tuesday, February 03, 2009

Noyantis TaskPanel 1.05

Version 1.05 of the Noyantis TaskPanel wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; Legacy compatibility added with Example app. The new version can be downloaded from the Members area using the original download and registration details contained in sales emails.

Posted Tuesday, February 03, 2009

Crisp Letter Icons

D-Icons has released its new Crisp Letter icon set. This is a set of letter and symbol icons based on a clean fresh pastel theme, to complement the d-crisp Icons set released a couple of weeks ago. There are 93 different designs, all in 12 vibrant colors at sizes from 128x128 down to 16x16. The symbols include the alphabet in upper and lower case, numbers, and most symbols available on a standard keyboard. The colors exactly match the d-crisp Icons.

Posted Tuesday, February 03, 2009

Noyantis CommandBars 1.14

Version 1.14 of the Noyantis CommandBars wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app (C55 Frame Extension to be completed); Legacy compatibility added with Example app; New methods added - 'SetComboSelection' & 'GetComboSelection'; Keybindings on Designed Bars could cause app to GPF if pressed. The new version can be downloaded from the Members area using the original download and registration details contained in sales emails.

Posted Tuesday, February 03, 2009

Clarion QuickBooks Connections Beta 0.91

Clarion QuickBooks Connections is a set of classes and templates that ease the access and update of QuickBooks data. The intro price is $149.

Posted Tuesday, February 03, 2009

SetupBuilder 6.9 Build 2482 Developer Edition Hotfix

This hotfix addresses a "#copy file..." compiler directive issue. To get the latest product version, select Help | Check for Updates... in your SetupBuilder IDE.

Posted Tuesday, January 20, 2009

Base64 Example

John Griffiths adapted some GPL (General Public License) C++ source for Base64 encoding and has incorporated it into a CW6.3 APP.

Posted Tuesday, January 20, 2009

SetupBuilder 6.9 Build 2481

Lindersoft has released SetupBuilder Version 6.9 Build 2481, the latest edition of its award-winning Installation Authoring and Configuration Management system for Microsoft Windows based applications. SetupBuilder Version 6.9 deploys applications to the Microsoft Windows Platform, including Windows 7, Windows Vista, Windows Server 2008 and Windows x64. This release is available, free of charge, to all SetupBuilder customers who have an active SetupBuilder maintenance subscription plan. This build brings SetupBuilder 6 in sync with SetupBuilder 7.

Posted Tuesday, January 20, 2009

Noyantis ShortcutBar 1.15

Version 1.15 of the Noyantis ShortcutBar wrapper template has been released. Modifications include: Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; BUG FIX: Legacy detection enhanced. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, January 20, 2009

J-HTML Improved

A new "Advanced" tab has been added to the J-Html control. At the moment this tab enables you to easily override hyperlink targets, and to set whether or not users can drop documents into the control. J-Html contains a huge amount of code that has never really been documented or made public. Typically this is code written for specific clients, or for internal use. The next version of J-Html will open up some of these features through this new "Advanced" tab.

Posted Tuesday, January 20, 2009

Noyantis PropertyGrid 1.03

Version 1.03 of the Noyantis PropertyGrid wrapper template has been released. Modifications include: Codejock ActiveX Auto Registration facility added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Tuesday, January 20, 2009

Noyantis CalendarPro 1.14

Version 1.14 of the Noyantis CalendarPro wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; Legacy compatibility added with Example app; New methods added - GetActiveStartDate & GetActiveEndDate; 2 new parameters added to NewDate_???? procedure (Active Start and End Date); New method added - SelectDayTime; New methods added - SetDayStartTime & SetDayEndTime; Restrict parameter added to SetDayLimits method; User definable conditions added to Right Click options; Override Office 2007 Theme facility added (includes over 15 new methods for altering appearance); Additional Embed points added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.
Posted Tuesday, January 20, 2009

Noyantis ShortcutBar 1.15

Version 1.15 of the Noyantis ShortcutBar wrapper template has been released. Modifications include: Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; BUG FIX: Legacy detection enhanced. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.
Posted Tuesday, January 20, 2009

Noyantis CalendarPro 1.14

Version 1.14 of the Noyantis CalendarPro wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; Codejock ActiveX Auto Registration facility added; C55 compatibility added with Example app; Legacy compatibility added with Example app; New methods added - GetActiveStartDate & GetActiveEndDate; 2 new parameters added to NewDate_???? procedure (Active Start and End Date); New method added - SelectDayTime; New methods added - SetDayStartTime & SetDayEndTime; Restrict parameter added to SetDayLimits method; User definable conditions added to Right Click options; Override Office 2007 Theme facility added (includes over 15 new methods for altering appearance); Additional Embed points added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.
Posted Tuesday, January 20, 2009

FinalStep 2.20

Changes in FinalStep 2.20 include changes needed in the template code for compatibility with Clarion 7 appgen. These changes may solve some previously unexplained occasional appgen GPFs in Clarion 6 as well.
Posted Tuesday, January 20, 2009

CPCS Addons For C7

All CPCS Addon products for C7.0 are now available for download from the CPCS website. Each of the addon product upgrades is free of charge to all existing registered users of any prior version of the same addon product. However, you will need to contact CPCS for new install codes. You will also need to have CPCS v7.00 installed as well. As with CPCS v7.00, these are Beta builds, and any issues encountered should be emailed directly to my attention rather than posting in these news groups.
Posted Tuesday, January 20, 2009

Clarion 6 UAC Fix 1.20

Version 1.20 of the "Clarion 6 UAC Installation Fix" application is now available. For compatibility reasons, it's still called "Vista fix". Windows Vista, Windows Server 2008 and Windows 7 provide a new security feature named User Account Control (UAC). An "UAC-aware" installer is required in order to correctly install your applications. Even if

your Clarion 3rd party vendor provides a UAC-aware installer, it's not possible to correctly auto-detect the installed Clarion version. The original Clarion 6 installer is unable to write to the WIN.INI file and to update the system path environment variable on Windows Vista, Windows Server 2008 and Windows 7. These actions require a UAC-aware installer with administrator execution level privileges. This freeware application makes the appropriate WIN.INI modifications so 3rd party installations can detect Clarion 6 on Vista, Windows Server 2008 and Windows 7 machines. It also adds the Clarion \Bin path to the system path environment variable. This is a freeware application and you can redistribute it to your customers. You can use the fix on Windows Vista, Windows Server 2008 and Windows 7. Because Clarion 6 cannot be run on 64-bit versions of Windows, the application detects x64 now and displays a warning. Just follow the wizard to apply the fix and all 3rd party installers should be able to auto-detect your Clarion 6 environment on Vista, Windows Server 2008, and Windows 7. Note: you don't need this fix for Clarion 7.
Posted Tuesday, January 20, 2009

CPCS v7.00 Beta

CPCS v7.00 Beta for Clarion 7 is now available for purchase from the CPCS website.
Posted Tuesday, January 20, 2009

Medical Icon Collection

1stLogoDesign announces the Medical Icon Collection. Due to the huge size the collection is being released in two stages. The first one is available now and it contains 10616 images in 10 sizes and ICO and PNG formats (around 233550 files, around 3.2gb download). The second will be released shortly. Sizes included: Sizes included: 16x16, 24x24, 32x32, 48x48, 64x64, 72x72, 80x80, 96x96, 128x128 & 256x256. Formats: Windows ICO, PNG, (Mac ICNS, PSD, JPG, BMP, and, GIF next release). Each icon includes 36 options: New, Add, Change, Delete, OK, Configure, Settings, Info, Warning, Left, Right, Up, Down, No Access, Search, Question, Refresh, Undo, Redo, Warning, Error, Secure, Unsecure, Lock, Unlock, Progress, Progress warning, Progress Error, and more. During the pre-release period the price is discounted by $100, and includes: First release of the collection 10616 images (now); Web 2.0 collection FREE (now); Second release of the collection (as soon as it is available).
Posted Tuesday, January 20, 2009

SetupBuilder Totorial

Stu Andrews has put up a walkthrough (tutorial) on using SetupBuilder to code-sign your own Exes/DLLs.
Posted Tuesday, January 20, 2009

Mallorca Lite Icons For $10

The Mallorca Lite Collection contains 178 unique icons, in the following sizes: 16x16, 24x24, 32x32, 48x48, 64x64, 72x72, 80x80, 96x96, 128x128, 256x256. File formats: Windows ICO, PSD, PNG, JPG, GIF, BMP, Mac ICNS. There are a total of 10,145 files (505.4 MB). These icons and images are royalty free.
Posted Tuesday, January 20, 2009

Noyantis PropertyGrid 1.01

Version 1.01 of the Noyantis PropertyGrid wrapper template has been released. Modifications include: Codejock Control Version selection enhanced; C55 compatibility added with Example app; Legacy compatibility added with Example app. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.
Posted Tuesday, January 20, 2009

Beta Test Web Site In Beta

BetaTestNow helps you to find testers for your software or test software that is listed on the site. The website itself is written (mostly) with Nettalk web server and Clarion 6.3. Register as company requiring testers for your software and search the database for suitable testers. You can list your software individualy and assign categories, renumeration types and keywords (eg:Clarion) to refine the type of tester that you wish to attract. Testers can be hired: For free; For a free license; For an hourly rate / fixed sum. You can also register as a tester and search the database for software to test, according to your preferences / keywords.

Posted Tuesday, January 20, 2009

Product Scope 7.5

Changes in Product Scope 7.5 include: Simpliefied user interface: Increased documentation; Longer folder support: Memo report; File search improvements; XP non-admin user support, and more. Introductory pricing is US$29.95 (regular $39.95).

Posted Tuesday, January 20, 2009

PDF-XChange Version 4

PDF-XChange Version 4 is now available for download. The template has not been tested against Clarion7. A C6/C7 installer will be released as soon as a definite schema is adopted and all tests are complete.

Posted Wednesday, January 07, 2009

CHT C7 Video Version 2

Gus Creces has provided a re-work of a 20 minute video in which he takes five CHT demo applications, convert the apps to C7 app format, and then compiles in C6 compiler mode and again in C7 compiler mode. The video is approx 80MB (1024x768).

Posted Wednesday, January 07, 2009

Noyantis ShortcutBar 1.14

Version 1.14 of the Noyantis ShortcutBar wrapper template has been released. Modifications include a bug fix to event triggering. The new version can be downloaded from the Members area using the original download and registration details contained in your sales emails.

Posted Wednesday, January 07, 2009

Clarion Third Party Profile Exchange January 2 2009 Release

An update to the Clarion Third Party Profile Exchange is available. This is a maintenance release, both online and data version. C7 Status was updated during this maintenance release - 122 products are marked as C7, 21 vendors included.

Posted Wednesday, January 07, 2009

Seven Icon Collections For $199

1st Logo Design is offering all of its current icon collections for $199 (a $723 value). This is a total of 15.9 Gb of images including: The Mallorca Collection; The Real Estate Collection; The Sterling Collection; The Rio Collection; The LNS Collection; The Webmaster Collection; The Web 2.0 Collection; The Background Collection (bonus set).

Posted Wednesday, January 07, 2009

Clarion Magazine

Proper Case in Clarion#

by Mike Hanson

Published 2009-01-16

Last month I wrote about a class I've long used to implement proper case in Clarion applications. After I completed that article I began to think about proper case functionality in Clarion#.

I have read that the hard part about learning .NET is not the new languages, but rather the massive library of components with which one must become familiar. With this in mind, I figured that there was probably an equivalent to my MHProperClass in the regular .NET library.

However, after searching around for a while, the closest I have found is the method TextInfo.ToTitleCase in the System.Globalization namespace. Not only does it have many of the shortcomings of Clarion's ENTRY,CAP approach, it doesn't currently handle strings which are passed as all uppercase. Even worse, its uppercase handling converts an "s" after an apostrophe to lowercase, even though it leaves the rest as all CAPS.

For example, when I pass in my test strings I get these results:

Passed	**Via ToTitleCase**	**Preferred**
mcdowell	Mc**d**owell	McDowell
john smith, md	John Smith, M**d**	John Smith, MD
james o'sullivan, phd	James O'**s**ullivan, Ph**d**	James O'Sullivan, PhD
andrew guidroz ii	Andrew Guidroz I**i**	Andrew Guidroz II
von richtofen	**V**on Richtofen	von Richtofen
alexander the great	Alexander **T**he Great	Alexander the Great
MCDONALD'S	M**CDONALD**'s	McDonald's
A PLACE NAMED "MCDONALD'S"	A P**LACE** N**AMED** "M**CD****ONALD**'s"	A Place Named McDonald's

If you search the net you'll discover that ToTitleCase is reviled by most .NET developers, and there are no obvious alternatives. There might be some other third-party component out there that does what MHProperClass can do, but I didn't find it. It's not worth any further searches for me, though, as I've already got the Clarion code. I just need to convert it to Clarion#, and how tough could that be?

Well... there are certainly a few obvious things that I need to consider: STRINGs versus CLASTRINGs, zero-based arrays, etc. I'll certainly address those. The biggest thing for me, though, is that even though Clarion# is a similar-looking language to Clarion, I need to approach Clarion# with a different mindset.

Variables aren't just variables anymore! They might be simple types, structures, or references to objects. You might protest that, except for .NET structures, we had those things before, and that's true. However, the issues weren't as in your face as they are now. I predict that this change in data types will present the biggest challenge to most Clarion developers as they move from C6/C7 to Clarion.NET.

So what should I do about STRINGs?

In Clarion#, a variable defined as a STRING is a reference to the .NET String class, not a good-old Clarion STRING. The String class doesn't support nifty features like slicing, which I use throughout my MHProperClass (although String does have a host of methods such as CopyTo, Insert, Replace, Split etc.). One of the biggest issues with .NET strings is that they're immutable; you can't manipulate a String directly, you can only create a new modified String.

I'll stay away from the String type for much of my class. This leaves me with two other obvious options:

- CLASTRING - This class represents an old style Clarion STRING. It supports all of the operations that I'm used to, and will probably require the fewest changes in my code. The key difference from C6/C7 for string slicing is that arrays in .NET are zero-based, whereas C6/C7 arrays start at 1.
- StringBuilder - This is a powerful class that permits many robust string management facilities and compensates for the immutability of Strings. However, it would require that I change all of my existing logic from slicing to the StringBuilder equivalents. Therefore, I'm probably going to skip it for this migration.

References front and center!

In .NET, only simple data types (e.g. LONG, BYTE) are "variables" in the old sense. These are called value types. Anything more complex is a class (a reference type), and you work with references to objects of those classes. Defining reference types in the data section is the first step; you must also instantiate them before you can use them.

The lines between value and reference types are often blurred. Strings, for instance, are reference types but are often treated in code the same as value types (more on that in a moment). Expect to stub your toes many times before you become completely comfortable with this new approach to the world.

Classes sometimes offer constructors that enable instantiation directly in the data section, while others do not. Let's look at CLASTRING for an example:

```
  PROGRAM
  NAMESPACE TestCLASTRING
  USING System
  MAP
  END
BlankVar        CLASTRING(10)      !10-char blank instance
InitializedVar  CLASTRING('Hello')  !5-char 'Hello' instance
BareRef         CLASTRING          !No instance
  CODE
  Console.WriteLine('>'& BlankVar &'<<')
  Console.WriteLine('>'& InitializedVar &'<<')
  BareRef = 'Some String'
  Console.WriteLine('>'& BareRef &'<<')
  Console.ReadKey
```

This code produces the following output:

```
>         <
>Hello<
```

```
>Some String<
```

Note that BlankVar and InitializedVar were both instantiated in the data section, while BareRef wasn't instantiated until the code section. If I had tried to access BareRef before doing this, a run-time exception would have occurred.

A word of warning: You cannot instantiate the String class in the data section the same way you instantiate a CLASTRING. It must be done in the code section. Also, most instantiations require that you use NEW *Classname*. Strings and CLASTRINGs are an exception to this rule, because they're used so often. Also, CLASTRING lets me set the string's value either way:

```
BareRef = 'Some String'
BareRef = NEW CLASTRING('Some String')
```

This is not an option for the String class. Try to NEW a String with a size or a string literal and you'll get a compiler error:

```
StringRef = 'Some String'            !Works
StringRef = NEW STRING('Some String')  !Fails
```

Confused yet? Just keep hammering away, and eventually it will become clear(er).

Revised class definition

The Clarion# version of MHProperClass looks quite similar to the Clarion version:

```
MHProperForcedWordQueue QUEUE,TYPE
Word                CLASTRING
PaddedUpper         CLASTRING
Length              LONG
NotAtStart          BYTE
                END

MHProperClass       CLASS,PUBLIC,NETCLASS
ForcedWord          MHProperForcedWordQueue,PRIVATE
Name                CLASTRING,PROTECTED
Length              LONG,PROTECTED

AcceptControl       PROCEDURE(SIGNED Feq),STRING,PROC
AddForcedWord       PROCEDURE(STRING Word,<BYTE NotAtStart>)
Construct           PROCEDURE
Destruct            PROCEDURE
DisposeForcedWord   PROCEDURE(LONG X),BYTE,PROC,PRIVATE
DisposeName         PROCEDURE,PROTECTED
HandleForcedWords   PROCEDURE,PROTECTED
ToProper            PROCEDURE(STRING Name),STRING
RemoveForcedWord    PROCEDURE(STRING Word),BYTE,PROC
RemoveForcedWords   PROCEDURE
                END
```

The only difference here is that &STRING has been changed to CLASTRING, which is also a simple reference.

NOTE: In early releases of Clarion.NET reference variables had to be prefixed with &, as in Clarion code. In most cases the leading & is no longer necessary. You can read more about this change in Should Clarion# Drop Automatic Instantiation? and Clarion# Array Index And Class Instantiation Changes.

Additionally, instead of declaring this class in an INC file and implementing it in a CLW file, I've placed both the class declaration and the class code in a CLN file. In Clarion you need at least the header file to use the compiled class; in Clarion#, as in other .NET languages, you only need a reference to the assembly (i.e. the DLL) containing the class; the .NET runtime will examine that assembly for accessible classes, methods, variables and properties.

Sometimes .NET Makes Life Easier

Here's how the AddForcedWord method looked in Clarion:

```
MHProperClass.AddForcedWord PROCEDURE(STRING Word,<BYTE NotAtStart>)
  CODE
  SELF.ForcedWord.Length       = LEN(CLIP(Word))

  SELF.ForcedWord.Word        &= NEW STRING(SELF.ForcedWord.Length)
  SELF.ForcedWord.Word         = Word

  SELF.ForcedWord.PaddedUpper &= NEW STRING(SELF.ForcedWord.Length+2)
  SELF.ForcedWord.PaddedUpper  = ' ' & UPPER(Word) & ' '

  SELF.ForcedWord.NotAtStart   = NotAtStart

  ADD(SELF.ForcedWord)
```

And here's how it looks in Clarion#:

```
MHProperClass.AddForcedWord  PROCEDURE(STRING Word,<BYTE NotAtStart>l;)
  CODE
  SELF.ForcedWord.Word        = CLIP(Word)
  SELF.ForcedWord.PaddedUpper = ' '&UPPER(SELF.ForcedWord.Word)&' '
  SELF.ForcedWord.Length      = LEN(SELF.ForcedWord.Word)
  SELF.ForcedWord.NotAtStart  = NotAtStart
  ADD(SELF.ForcedWord)
```

Now that both allocation and initialization are performed as part of the reference assignment, resulting in fewer lines of code.

References in queues

With C6/C7, a reference in a queue record must be disposed of before the queue record is deleted, which is why the old C6/C7 DisposeForcedWord method looked like this:

```
MHProperClass.DisposeForcedWord PROCEDURE(LONG X)!,BYTE
  CODE
  GET(SELF.ForcedWord, X)
```

```
IF ERRORCODE() = 0
  DISPOSE(SELF.ForcedWord.Word)
  DISPOSE(SELF.ForcedWord.PaddedUpper)
  DELETE(SELF.ForcedWord)
  RETURN 0
ELSE
  RETURN 1
END
```

In .NET, the garbage collector is supposed to handle the disposition of any object that's no longer referenced (although it might take a while). I've asked for clarification from SoftVelocity whether this is an issue for queues, especially if we don't want to wait for the garbage collector to clean it up. The new AUTODISPOSE attribute is helpful for regular reference variables, but I don't think it would work with QUEUE fields. For now, my Clarion# version of the method looks like this:

```
MHProperClass.DisposeForcedWord  PROCEDURE(LONG X)!,BYTE
 CODE
 GET(SELF.ForcedWord, X)
 IF ERRORCODE() = 0
  DISPOSE(SELF.ForcedWord.Word)
  SELF.ForcedWord.Word = NULL
  DISPOSE(SELF.ForcedWord.PaddedUpper)
  SELF.ForcedWord.PaddedUpper = NULL
  DELETE(SELF.ForcedWord)
  RETURN 0
 ELSE
  RETURN 1
 END
```

The DISPOSE and NULL assignments are probably overkill, and it may be that I can delete those lines and leave everything up to the garbage collector. There's nothing inherently wrong with this approach for now, though. Once I get more information I may change it.

Zero-based arrays

I'm not going to argue the relative merits of 0-based versus 1-based arrays. In the .NET world things tend to be 0-based, and SoftVelocity recognized (after a bit of cajoling) that it was best to have Clarion# use this same convention. This means better interoperability between Clarion# and other .NET languages. Unfortunately, it does mean that some of my existing code will need to change as I move it to Clarion#.

A good example of this is the HandleForcedWords method, which first modifies a copy of the Name so that all non-alphanumeric characters are converted to spaces. In Clarion, the LOOP bounds were:

```
LOOP X = SELF.Length+1 TO 2 BY -1
```

In Clarion#, this changes to:

```
LOOP X = SELF.Length TO 1 BY -1
```

As expected, everything's adjusted down by 1 to accommodate the shifted array numbering scheme. Similar changes are made in the ToProper method, including a comparison between the current character position and the length of the string.

INSTRING's twist

In Clarion, INSTRING returns the starting position of the found substring. If the substring is not found, then INSTRING returns zero. This presents a problem in Clarion#, where 0 is a valid character position. SoftVelocity elected to change the behavior and documentation of INSTRING slightly to accommodate this. Instead of returning the starting position of the substring within the larger string, it returns the starting "step". (This is related to the step-size passed to INSTRING, which is 1 for most situations.) Therefore, if the substring coincides with the start of the larger string the return value will still be 1, even in .NET. That means I need to subtract 1 from INSTRING's result before using that value as a position indicator.

I also have to worry about the starting position passed into INSTRING, which must use the 0-based array numbering scheme. Here's the original Clarion code:

```
P = CHOOSE(~SELF.ForcedWord.NotAtStart, 1, 2)
LOOP
  P = INSTRING(SELF.ForcedWord.PaddedUpper, N, 1, P)
  IF P = 0 THEN BREAK.
  SELF.Name[P : P+SELF.ForcedWord.Length-1] = SELF.ForcedWord.Word
  P += 1
END
```

Here's the Clarion# version:

```
P = CHOOSE(~SELF.ForcedWord.NotAtStart, 0, 1)
LOOP
  P = INSTRING(SELF.ForcedWord.PaddedUpper, N, 1, P)
  IF P = 0 THEN BREAK.
  SELF.Name[P-1 : P+SELF.ForcedWord.Length-2] = SELF.ForcedWord.Word
  !P += 1
END
```

Things to note:

- The initial starting position, defined in the first line, is 0 or 1, not 1 or 2.
- INSTRING continues to return 0 if the search fails.
- The return value from INSTRING must be reduced by 1 to be used as a position within the string.
- Since the value returned by INSTRING is 1 larger than the actual starting position, I don't need to increment the starting position at the bottom of the loop.

Deferred functionality

Recall that the Clarion MHProperClass has an AcceptControl method for changing a control's contents at runtime. I could do a quick and dirty implementation of this for .NET, but I just realized that Clarion's own Clarion.Windows.Forms.Entry class doesn't even support CAP yet. (I'm not sure whether this is a bug or a purposeful omission.) Therefore, I've decided I would rather create a derived ENTRY class that forces the ToProper conversion as the name is entered. Unfortunately, that's beyond the scope of this article.

Source files

The downloadable Source.zip contains the following files:

- TestProper.sln - Test solution
- TestProper.cnproj - Test project
- TestProper.cln - Main test source
- MHProper.cln - MHProperClass source

Conclusion

After I got over the Clarion# learning curve, the substantive changes I made to the code were minimal: I converted the &STRING references to CLASTRING and adjusted my string splicing for 0-based arrays. Of course, MHProperClass was already object-oriented code, which made the job much easier. Procedural code would require significantly more rethinking.

Download the source

Mike Hanson is affiliated with BoxSoft, which produces the "Super" series of templates, distributed through Mitten Software. He has been creating add-on products for Clarion since his Public Domain Models for CPD 2.0 back in 1988. He's also written articles for every Clarion-related publication, and has spoken at numerous conferences and training seminars. If you have any questions, you can reach him via www.boxsoft.net.

Reader Comments

Add a comment

Clarion Magazine

Buying A Code-Signing Certificate: The 2008 Version

by Jane Fleming

Published 2009-01-30

When we old people talk about the "good old days," we tend to wax nostalgic about walking 12 miles through the snow to school and back - uphill each way. Well, with similarly rose-colored glasses I look back fondly at my first code-signing certificate purchase experience in 2006. At the time, I wrote a couple of Clarion Magazine articles about the purpose and mechanics of code-signing and how to negotiate the certificate purchase through the Comodo website (Comodo having been much less expensive than Verisign).

My two-year certificate was expiring so it was time to get a new one. But in that Vista (and Server 2008) put so much emphasis on signed applications, greed seems to have reared its head within the certification companies. My 2-year Comodo certificate had cost $179 in 2006. Now the same 2-year certificate from Comodo costs $339 and a 3-year certificate costs $500. Ouch! That's still a "bargain" I suppose, as Verisign's price is now $499 for one year and $1,293 for 3 years.

Fortunately, being a user of Lindersoft's SetupBuilder product I have the ability to purchase Comodo certificates through Lindersoft rather than through Comodo's "front door" - and an identical 3-year Comodo certificate costs $200 via this route. This deal is also available to those who have a Lindersoft "Community Membership" rather than a SetupBuilder license. Needless to say, taking advantage of the deal Lindersoft has negotiated was a no-brainer.

The threads I came across on Friedrich's and on the Clarion news groups indicated that it was best to start fresh with Comodo for this type of purchase, rather than using the same account used for a previous purchase. So I created a new email address for this purpose. (I used my regular address as the contact info on the certificate itself.) And, credit-card in hand, I went online to buy a new certificate.

I had also been reading a certain number of horror stories in the past year, and had learned from those that the ActiveX control used for issuing certificates will not work on a Vista computer. Be clear on this - Yes, the certificate (once you have it in your hot little hands) *will* work on a Vista machine. And yes, the same signtool.exe that I downloaded as explained in the 2006 articles runs just fine to sign stuff on my 64-bit Vista machine. But No, you can't use a Vista machine to make the purchase and download. And you do need to use Internet Explorer, not another browser.

But just in case you're tempted to try Vista, here's the initial Comodo screen when accessed through Internet Explorer on a Vista machine (Figure 1).

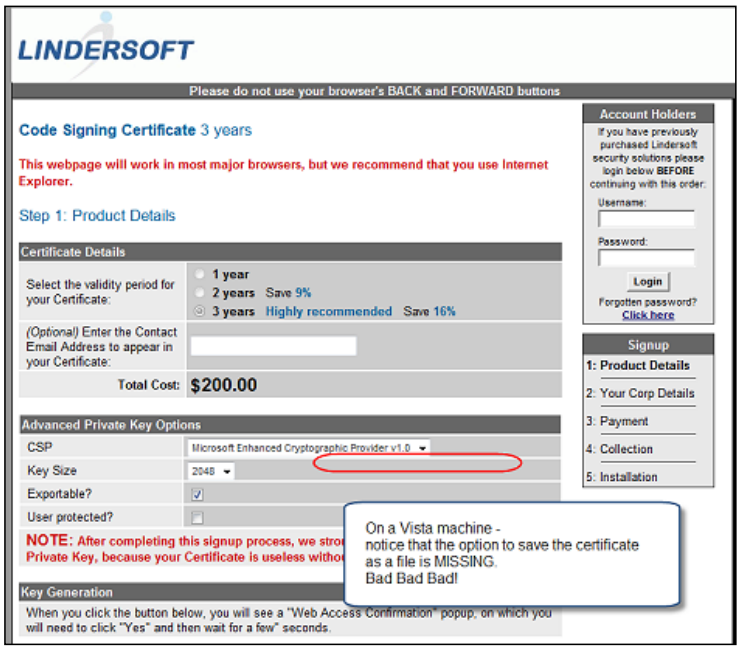


Figure 1. On a Vista machine, where to store the certificate is missing!

Figure 2 shows the same screen through an XP machine. Notice that the important information of where to store your private key file and certificate are missing on the Vista machine. You want those in files that you can back up and can use on any development machine you may have.

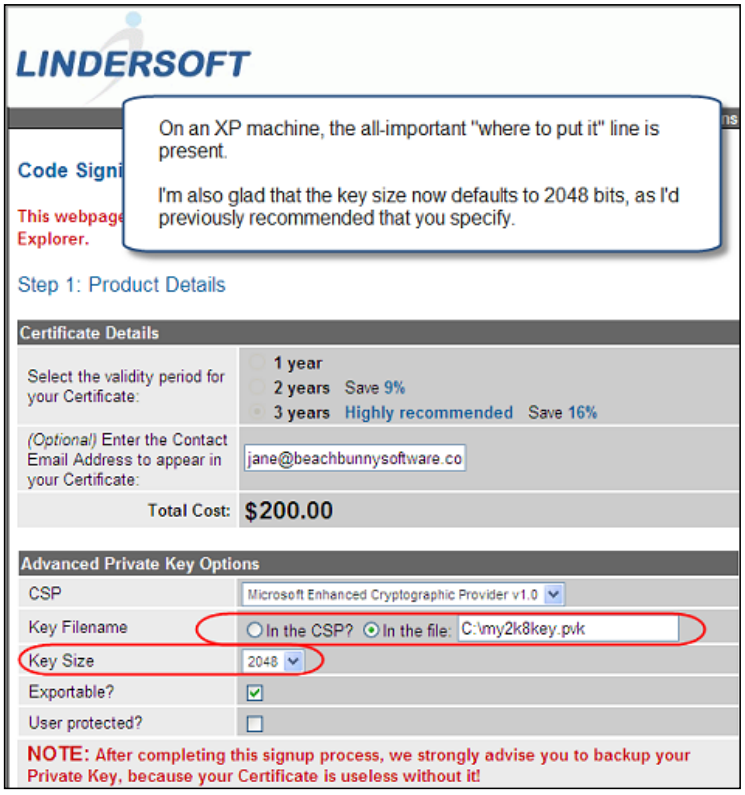


Figure 2. Purchase screen on an XP machine.

Well, my laptop purchase this year was a Sager with a 64-bit Intel processor. I bought it with no O/S installed and actually deliberately installed 64-bit Vista. I still see a lot of trash talk about Vista. I personally feel that having day-to-day immersion in a work environment similar to that used by increasing numbers of my users is a worthwhile reality

check. Having full access to the laptop's 4 gigs of RAM is a nice thing, I've found this O/S to be quite stable, with one notable exception. Strangely, if I use the scroll wheel on my mouse while running Acrobat Pro 9.0, I can rather consistently BSOD my machine, so I've mostly switched to PDFTools.

I knew in making the decision to go with 64-bit Vista that I'd need to make some provision for running the 16-bit Clarion 6 IDE. I went with VMWare, in part due to its support for USB devices in the client machines.

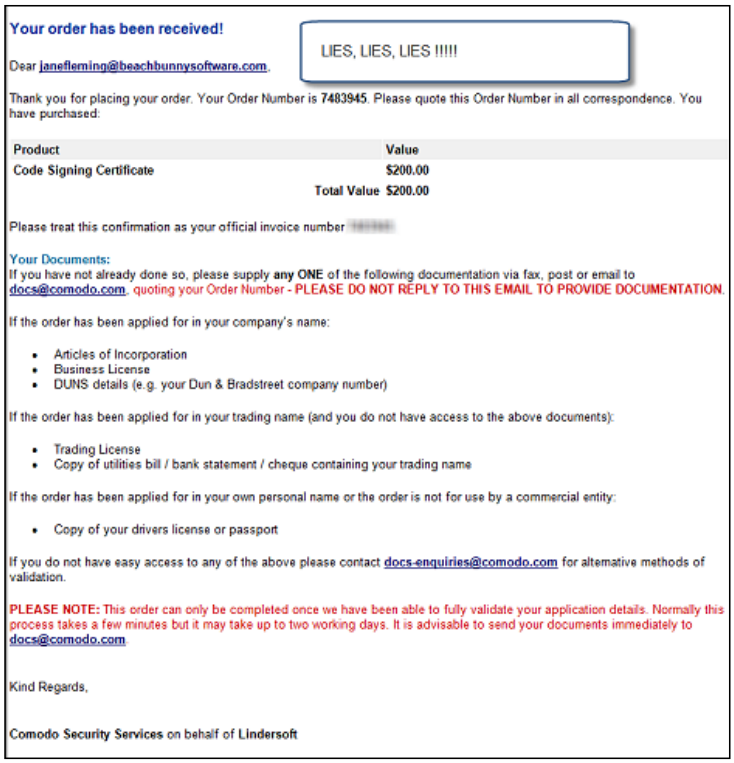
So having 32-bit XP running in a VMWare machine, I decided to use that to purchase my certificate.

Well, Señor Murphy wound up teaching me yet another lesson... not the least of which is not to be overly confident. (Actually, he's been trying to teach me that for years...)

I went through the purchase process. As you see in Figure 2, I saved the private key file to my virtual C drive. (And then backed it up onto a flash drive.)

Shortly after sending my money, I received an email from Comodo (Figure 3) similar to the one I'd gotten during my 2006 purchase: email them a document to prove I am who I say I am.

Last time, I'd sent a PDF scan of my city business license. And so I did that again.



Your order has been received!

LIES, LIES, LIES !!!!!

Dear janefleming@beachbunnysoftware.com,

Thank you for placing your order. Your Order Number is **7483945**. Please quote this Order Number in all correspondence. You have purchased:

Product	Value
Code Signing Certificate	$200.00
Total Value	$200.00

Please treat this confirmation as your official invoice number [redacted]

Your Documents:
If you have not already done so, please supply **any ONE** of the following documentation via fax, post or email to docs@comodo.com, quoting your Order Number - PLEASE DO NOT REPLY TO THIS EMAIL TO PROVIDE DOCUMENTATION.

If the order has been applied for in your company's name:

- Articles of Incorporation
- Business License
- DUNS details (e.g. your Dun & Bradstreet company number)

If the order has been applied for in your trading name (and you do not have access to the above documents):

- Trading License
- Copy of utilities bill / bank statement / cheque containing your trading name

If the order has been applied for in your own personal name or the order is not for use by a commercial entity:

- Copy of your drivers license or passport

If you do not have easy access to any of the above please contact docs-enquiries@comodo.com for alternative methods of validation.

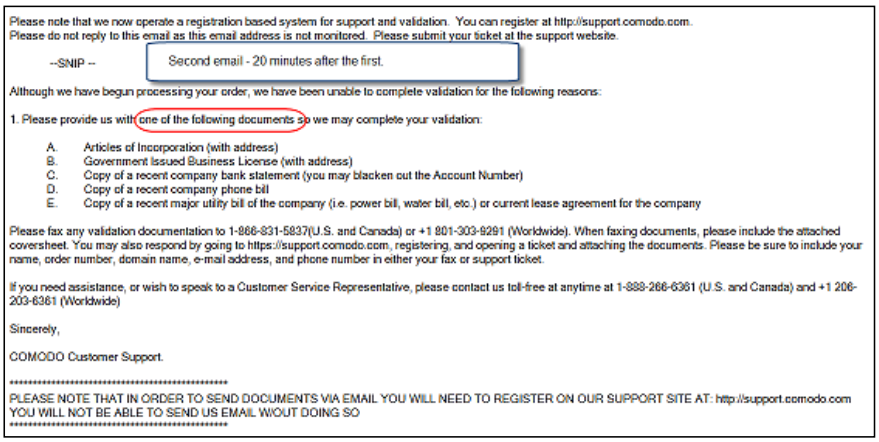
PLEASE NOTE: This order can only be completed once we have been able to fully validate your application details. Normally this process takes a few minutes but it may take up to two working days. It is advisable to send your documents immediately to docs@comodo.com.

Kind Regards,

Comodo Security Services on behalf of **Lindersoft**

Figure 3. No, it's not really going to be THIS easy!

Twenty minutes later, I received another email (Figure 4) again asking for *one piece* of documentation. But this time, rather than saying it could be sent to the docs@comodo.com address indicated in the first email, I need to register on their support site if I don't want to FAX the document.


Please note that we now operate a registration based system for support and validation. You can register at http://support.comodo.com.
Please do not reply to this email as this email address is not monitored. Please submit your ticket at the support website.

--SNIP --

Second email - 20 minutes after the first.

Although we have begun processing your order, we have been unable to complete validation for the following reasons:

1. Please provide us with one of the following documents so we may complete your validation:

A. Articles of Incorporation (with address)
B. Government Issued Business License (with address)
C. Copy of a recent company bank statement (you may blacken out the Account Number)
D. Copy of a recent company phone bill
E. Copy of a recent major utility bill of the company (i.e. power bill, water bill, etc.) or current lease agreement for the company

Please fax any validation documentation to 1-866-831-5837(U.S. and Canada) or +1 801-303-9291 (Worldwide). When faxing documents, please include the attached coversheet. You may also respond by going to https://support.comodo.com, registering, and opening a ticket and attaching the documents. Please be sure to include your name, order number, domain name, e-mail address, and phone number in either your fax or support ticket.

If you need assistance, or wish to speak to a Customer Service Representative, please contact us toll-free at anytime at 1-888-266-6361 (U.S. and Canada) and +1 206-203-6361 (Worldwide)

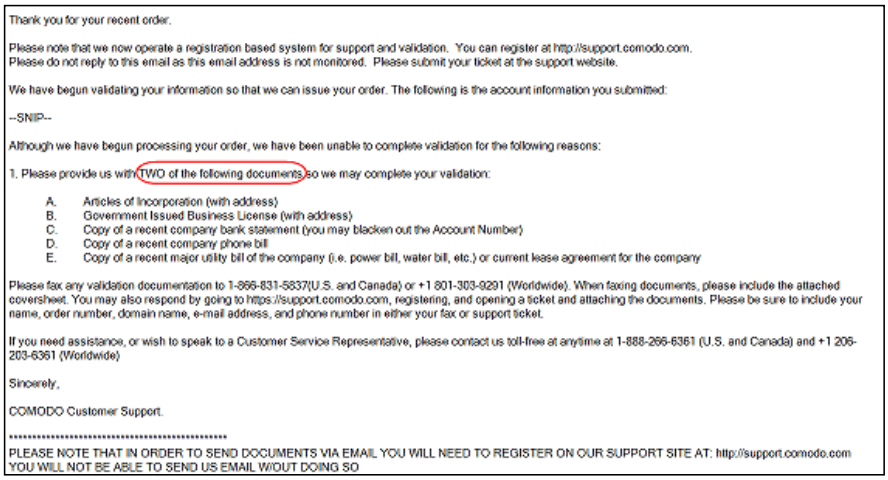
Sincerely,

COMODO Customer Support.

****
PLEASE NOTE THAT IN ORDER TO SEND DOCUMENTS VIA EMAIL YOU WILL NEED TO REGISTER ON OUR SUPPORT SITE AT: http://support.comodo.com
YOU WILL NOT BE ABLE TO SEND US EMAIL W/OUT DOING SO
****

Figure 4. Second email.

And one minute later, yet another email requiring *two* items of documentation (Figure 5).


Thank you for your recent order.

Please note that we now operate a registration based system for support and validation. You can register at http://support.comodo.com.
Please do not reply to this email as this email address is not monitored. Please submit your ticket at the support website.

We have begun validating your information so that we can issue your order. The following is the account information you submitted:

--SNIP--

Although we have begun processing your order, we have been unable to complete validation for the following reasons:

1. Please provide us with TWO of the following documents so we may complete your validation:

A. Articles of Incorporation (with address)
B. Government Issued Business License (with address)
C. Copy of a recent company bank statement (you may blacken out the Account Number)
D. Copy of a recent company phone bill
E. Copy of a recent major utility bill of the company (i.e. power bill, water bill, etc.) or current lease agreement for the company

Please fax any validation documentation to 1-866-831-5837(U.S. and Canada) or +1 801-303-9291 (Worldwide). When faxing documents, please include the attached coversheet. You may also respond by going to https://support.comodo.com, registering, and opening a ticket and attaching the documents. Please be sure to include your name, order number, domain name, e-mail address, and phone number in either your fax or support ticket.

If you need assistance, or wish to speak to a Customer Service Representative, please contact us toll-free at anytime at 1-888-266-6361 (U.S. and Canada) and +1 206-203-6361 (Worldwide)

Sincerely,

COMODO Customer Support.

****
PLEASE NOTE THAT IN ORDER TO SEND DOCUMENTS VIA EMAIL YOU WILL NEED TO REGISTER ON OUR SUPPORT SITE AT: http://support.comodo.com
YOU WILL NOT BE ABLE TO SEND US EMAIL W/OUT DOING SO

Figure 5. "We meant TWO items!"

I don't have a separate utility bill in my business name. So I scanned the header of a business bank statement to PDF to go along with the business license PDF and went to Comodo's support website. It's fairly straightforward to open a ticket. After you log in, choose the Certificates option in the Select Product dropdown and then you'll arrive at this screen (Figure 6). I was able to attach my PDFs and submit the ticket.

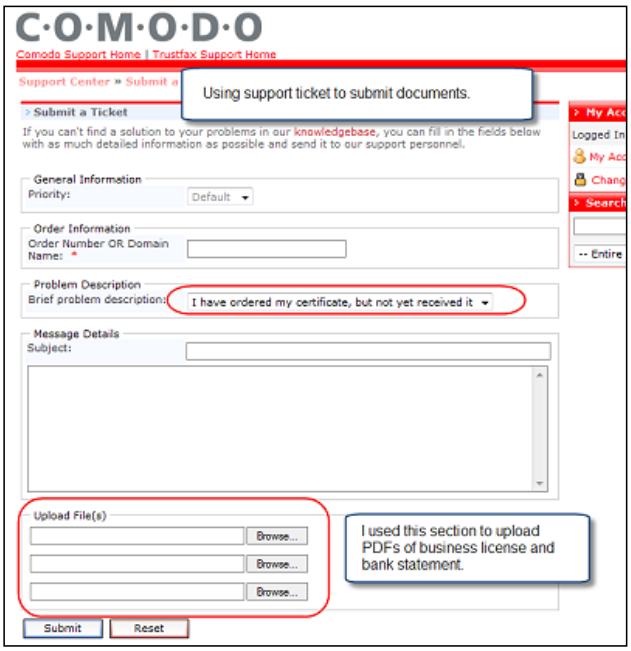


Figure 6. Submitting support ticket.

The support wizard confirmed receipt of my ticket (Figure 7) and I figured I'd earned a glass of cabernet.

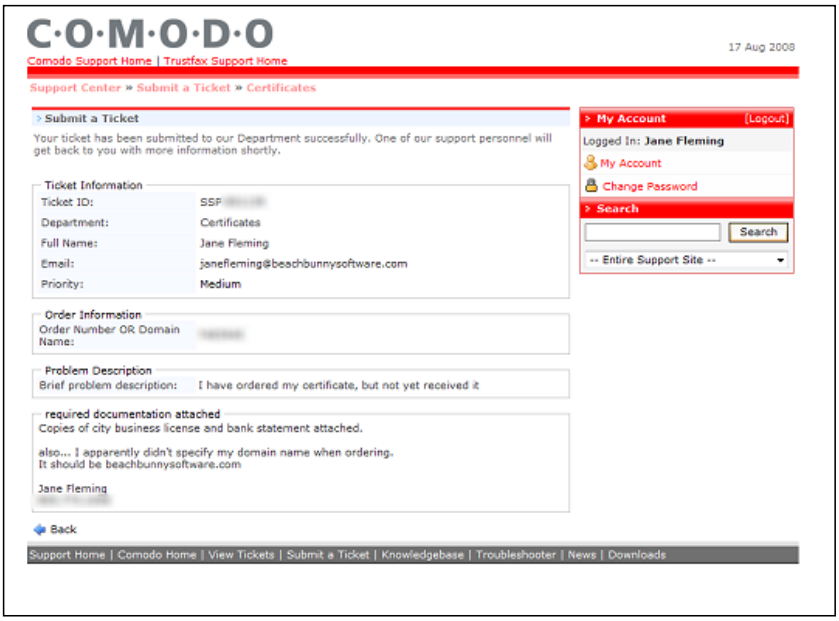
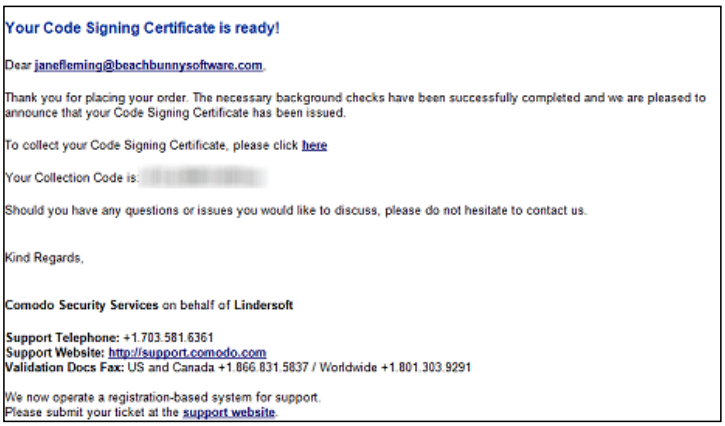


Figure 7. Ticket confirmed.

I'd begun the order process on a Saturday evening, and didn't expect anything to transpire until at least Monday. Monday morning, I received an email with information on collecting my certificate (Figure 8).



Your Code Signing Certificate is ready!

Dear janefleming@beachbunnysoftware.com.

Thank you for placing your order. The necessary background checks have been successfully completed and we are pleased to announce that your Code Signing Certificate has been issued.

To collect your Code Signing Certificate, please click here

Your Collection Code is:

Should you have any questions or issues you would like to discuss, please do not hesitate to contact us.

Kind Regards,

Comodo Security Services on behalf of **Lindersoft**

Support Telephone: +1.703.581.6361
Support Website: http://support.comodo.com
Validation Docs Fax: US and Canada +1.866.831.5837 / Worldwide +1.801.303.9291

We now operate a registration-based system for support.
Please submit your ticket at the support website.

Figure 8. Email with info on collecting the certificate.

Well, that wasn't too bad. Right?

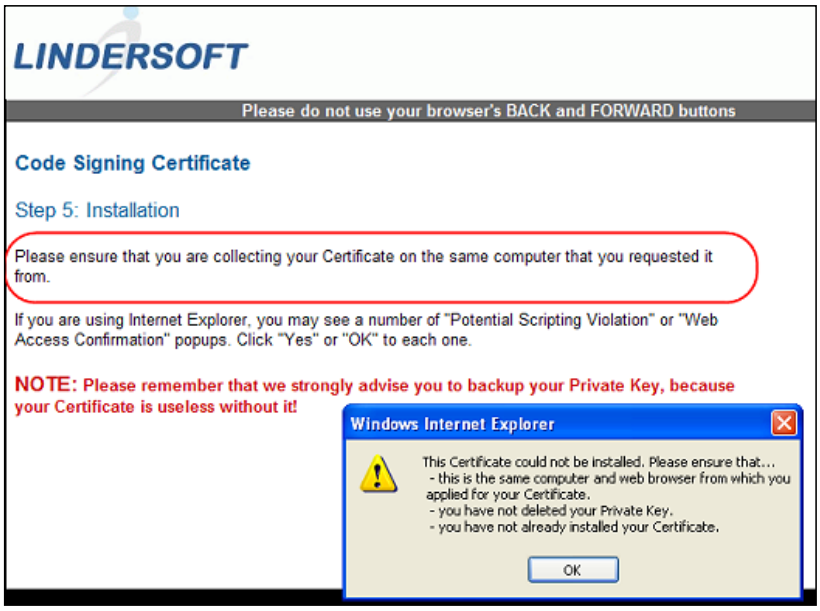
Did I mention the beloved Monsieur Murpheee ??

Somehow over the weekend I managed to crash my XP virtual machine (it was open when the aforementioned Acrobat provoked a Blue Screen of Death on my host machine). I now got an lsass.exe error when I tried to log on.

I had backup snapshots of the VM, but none taken since I'd made the certificate request. But... I'd backed up the private key file (and its password) so I should be good to go.

"Should" has always been my least favorite word in the English language.

I thought the easiest thing would be to use another XP machine to download the certificate. Well, obviously the ordering process stashes other information, because the collection didn't work (Figure 9)



LINDERSOFT

Please do not use your browser's BACK and FORWARD buttons

Code Signing Certificate

Step 5: Installation

Please ensure that you are collecting your Certificate on the same computer that you requested it from.

If you are using Internet Explorer, you may see a number of "Potential Scripting Violation" or "Web Access Confirmation" popups. Click "Yes" or "OK" to each one.

NOTE: Please remember that we strongly advise you to backup your Private Key, because your Certificate is useless without it!

Windows Internet Explorer

This Certificate could not be installed. Please ensure that...
- this is the same computer and web browser from which you applied for your Certificate.
- you have not deleted your Private Key.
- you have not already installed your Certificate.

OK

Figure 9. Curse you, Mr. Murphy!!!

Argh!

Well, I decided to see how lucky I might get.

I pointed my VMWare machine's CD drive at an .iso image of an XP installation CD and told its BIOS to boot from CD. When the Install Windows wizard got through initializing, I opted to repair an existing installation. After 30 or 40 minutes of cogitation it announced that it was finished.

And lo, I could boot into my XP virtual machine again.

And whatever info had been stashed (in the Registry or wherever) hadn't been destroyed.

And I downloaded my certificate, made two gazillion backup copies of it and the private key, and lived happily ever after.

Bottom line suggestions:

1. Read my previous code-signing articles.
2. Use the Lindersoft "deal" to buy your certificate. (Make sure you buy a code-signing certificate, not an SSL certificate.)
3. If you've previously ordered from Comodo, create a new email address for ordering through Lindersoft.
4. Be aware of what documentation is going to be required.
5. Use an XP machine and Internet Explorer for the ordering process.
6. Opt for 2048-bit key size, and to save your certificate in a file rather than in the cryptographic service provider store on the machine.
7. Be aware that you'll need to use THE SAME XP MACHINE to collect your certificate. (Afterwards, you can copy the certificate and private key files as you wish.)

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

Clarion Magazine

Managing Global Stuff With The Datafier

by Jeffrey Slarve

Published 2009-01-29

It's not very often that a life-changing idea happens. I mean something that really changes the way that you do things. Sometimes it's something that can be difficult to grasp, like the invention of the telephone, nuclear power, the Clarion template language, or puberty.

Other times, it's something stupidly simple. Like those little bumpy things on the freeway to keep you from veering off the road, coat hangers, Velcro, or puberty.

Well, luckily, this one is really simple. So simple, in fact, that it's silly. But I think it's really cool, too.

How this came about

Everyone who's written a threaded application has had the fun task of dealing with at least some global data. If a variable needed to be available globally in a multi-dll app, there was the further complication of having to first declare the variable, then export it, then re-compile all the applications to make use of it. That can be a huge pain in the backside and an enormous time stealer. Plus, with the advent of Clarion 6, global data got trickier. And time is money.

Geez. All I want to do was store a value. Why do I have to do all of this stuff? I just wanted a thread safe way to stick global data somewhere without having to go through all of the grief every time.

So I started simple

At first, I wrote a simple class to manage a queue full of variables. Basically this was a queue with two fields - one for the label, and one for the data. The data field was just a string. And all was good. I had a couple of methods for Setting and Getting the data. It worked okay, but I really didn't see the potential of what could be done, so I left it alone for a while.

And the features crept in

My class wasn't really thread safe. It was simple (I like simple), but I couldn't really feel comfortable using it as-is in a multi-threaded app, so I went ahead and put it back on the drawing board. I still wanted it to be as simple and easy to use as possible. I also wished that I could do a few other things with my data, such as actually using it. Encapsulation is great and all, but if it's too hard to get at the data then I might as well declare some variables and move on. There are also times when I want to actually specify what data type my data is going to live in. ANY variables (the data type "ANY") are often just fine, but if I need a DECIMAL(21,2), then it shouldn't be a big deal to create that. Then I thought it would be cool if I could store expressions (such as Format(Today(),@d2). And what about binding those labels so I could use them with my own Evaluate() outside of my class? My simple little class was starting to get a little bit less simple. Oh yeah, how about OptionExplicit (Visual Basic-speak for forcing a declaration). And namespaces, too. I think it would be cool to add a validation mechanism, along with event based notifications that broadcast to other threads that something's happened to the data. And how about automatically encrypting data as it gets stored? And what about exporting to XML?

But not this time.

I finally decided to just stop with the features, or I'd never write this thing. So for now, I left out a few new things until

version 2.

Enter the DataFier

The DataFierClass is what crawled off my drawing board. Spawned by inconvenience and laziness, it was forged from the nethers of the encapsulated darkness of various metaphoric descriptions that have yet to be determined.

The Datafier is basically just a place to put your data. It's kind of a server for variables. All you have to do is give your data a name, and Datafier will create a slot for it:

```
JSDF:SetValue('YourName','Frankopotomus')
```

You can retrieve that value with something like this:

```
Message(JSDF:GetValue('YourName'))
```

To get the downers over with, I might as well mention the drawbacks to this approach. As many of us already know, this kind of thing could lead to mis-spellings just like you would face when using implicit variables. And, most certainly, the setting/getting of values by a string key label can't be all that efficient. So it's a give and take. You give up a little efficiency for a lot of convenience. In addition to that, datafier variables can't easily be USE() variables on windows/reports. They can be used like that, but that's not the kind of use for which the Datafier was designed.

On the other hand, if you have a multi-dll app, you don't have to declare anything extra or recompile any other dll in order to have the data available to all. You also don't have to worry about thread safety because that's built in.

You can specify a datatype and a picture token, if desired. Something like this:

```
JSDF:SetValue('TodaysDate',Today(),'@d2',DataType:Long)
```

Notice the Datatype:Long equate. That comes from EQUATES.CLW in your libsrc folder. You can pass pretty much any of the datatypes listed (where applicable), and the Datafier will be sure to create that type. For most uses, however, it's probably just as well to omit the datatype and let Datafier use the ANY datatype by default.

The '@d2' that's passed is a picture token. Once you've specified the token, you can pass a True flag as the second parameter to get back formatted data. So this code:

```
JSDF:GetValue('TodaysDate',TRUE)
```

returns the formatted date.

If you specify, for example, a DataType:Decimal, you need to also let the Datafier know the extra info about it. Say, you want a Decimal(21,2). You'll specify the length and places this way:

```
JSDF:SetValue('WidgetFactor',1200.32,,DataType:Decimal,21,2)
```

That's nice, but how about binding and namespaces?

The data that you create with the datafier is optionally bindable. Meaning, you can bind the variables for use in Evaluate (), PROP:Filter, and wherever else you need to use a bound variable. First, you need to create a variable, using the bindable parameter:

```
JSDF:SetValue('WidgetFactor',1200.32,,DataType:Decimal,21,2,TRUE)
```

```
JSDF:SetValue('Coolness',100.29,,,,,TRUE)
```

Okay, now that I have a couple of variables that were set as bindable, I bind:

```
JSDF:BindData
```

Then I can make use of my newly bound stuff.

```
Message(Evaluate('WidgetFactor * Coolness'))
JSDF:UnbindData
```

Note that you'll probably want to avoid this kind of thing if you think it's possible that more than one thread will use that data at a time. Since you're accessing the data in a non-encapsulated way, that can't be terribly thread safe. But it's easy to do, if that's what you want. It's also part of the reason that I added Namespaces.

Namespaces, as far as the Datafier is concerned, provide an extra degree of separation and grouping of your data. Think of it as globally accessible procedure data, or perhaps the groupings that you can do in an ini file (the stuff in the brackets, e. g. [WINDOW]). If you make a unique namespace for a particular place in a particular thread, then I think it's pretty dang safe to bind and make use of those Datafied variables.

Like the variable name, the namespace is just a string that you create.

```
JSDF:SetValue('WidgetFactor',1200.32,,DataType:Decimal,21,2,TRUE,'Thread' & Thread())
JSDF:SetValue('Coolness',100.29,,,,,TRUE,'Thread' & Thread())
```

Here I've placed the WidgetFactor and Coolness variables in the 'Thread1' (or whatever the thread number was) namespace. You can make the namespaces as unique as they need to be; just respect the namespaces you've set up and all will be well.

Where applicable, the Datafier methods include a namespace parameter. So if you have hundreds or thousands of variables, but only want to bind a few of them, just group them by namespace and bind that namespace:

```
JSDF:BindData('MySpecialNamespace')
```

But where's the OOP?

I keep talking about the Datafier class, but my examples have thus far been straight procedure calls. The reason for that, is that instead of trying to manage a global object across a multiple DLL application, I just keep it inside one DLL. From that DLL I export some wrapper functions that are placed inside the global map of each DLL. The only place that needs to concern itself with the class is the one DLL that wraps the class.

But if you really want to use the class locally, then that's certainly an option. All you have to do is instantiate an object like this:

```
DFC  DataFierClass
```

So what can I do with all of this?

I keep having idea after idea about what to do with this stuff. Here are a couple of possible applications. What if, for example, you want to do some simple grid calculation, but don't want to fire up Microsoft Excel. Just imagine what a grid looks like. You have X (columns) and Y (rows). As long as your naming supports that metaphor, you can set/get the data:

```
X Long !I could have used the datafier to declare these, but meh.
Y Long
  Code
  Loop X = 1 to 100 ! 100 virtual columns
     Loop Y = 1 to 100 !100 virtual rows
      JSDF:SetValue(x & '.' & y, x * y)
     end
  end
```

I've created the variable labels by concatenating the X and Y values. So basically I'm doing (admittedly less efficient) array processing on the fly. I've created a virtual grid of 100 by 100 (10,000 squares). (The x * y part of it was just to give each "cell" value; the data could have been anything.

Okay. So I have a grid. What if I wanted to tally up the values of columns 2 and 3? Remember, the virtual columns are the X variable. Since I'm in charge of naming, I'll just use the row and column names to retrieve the data that I created.

```
  Loop X = 2 to 3 !Traverse columns 2 and 3
   Loop Y = 1 to 100 !100 virtual rows
    w += JSDF:GetValue(x & '.' & y)
   end
  end
```

Bam. That's it.

Here's another use, albeit a little more advanced. What if you wanted to be able to make API calls without having to declare a CString?

You could write your own wrapper for the DataFier that created a CString from a passed string literal.

```
NewCString  Procedure(String pData)!,*CSTRING
CS &CString
Name CString(60)
  Code
  Name = Clock()+Random(0,0FFFFh)!or whatever you want to do for a unique label.
  DFC.SetValue(Name,pData,,DataType:CString,Len(pData)+1)
  CS &= DFC.GetAddress(Name)
  Return CS
```

If you wanted to get fancy, you could utilize a namespace called "Throwaway" or something like that. Then you could write a garbage collection function to Dispose of the throwaway variables all at once.

What if I prefer to force that my variables get declared, vs suffering through a mis-spelling hell?

There is Visual Basic-like property in the Datafier class called OptionExplicit, which requires you to explicitly declare variables. In cases where OptionExplicit is turned on, you are forced to use the Declare() method. If you call SetValue () with a label that was not previously declared, then Datafier will return a JSDF:Notify message. In most cases I leave OptionExplicit turned off.

Another thing that you could do to enforce consistent naming is use equates for the labels. You'd just need to ensure that

the equates were available throughout your apps.

```
eSpecialCode  EQUATE('SpecialCode')
  Code
  JSDF:SetValue(eSpecialCode,'Top Secret Info Here')
```

This is still an improvement over shared global variables because you don't need to recompile any other DLLs in order to use a new equate.

Expressions

There is an special datatype equate that lets you use expressions with the Datafier:

```
DataType:JSDFExpression EQUATE(255)
```

Here's an expression that creates a timestamp:

```
JSDF:SetValue('TimeStamp',|
        'Clip(Left(Format(Today(),@d2.))) & " " |
        & Clip(Left(Format(Clock(),@t3.)))',|
        ,DataType:JSDFExpression)
```

To retrieve your 'TimeStamp', just do a JSDF:GetValue('TimeStamp'). Bam.

Pseudo read-only equates

The Datafier allows you to flag a variable as read-only, which turns the variable into a sort of equate:

```
JSDF:SetValue('ID','12344321A2124321')
JSDF:SetReadOnly('ID',,TRUE)
```

Note: Although you can't subsequently modify the value within the Datafier, the ReadOnly attribute has no effect if you retrieve the address of the variable (with the GetAddress method) and manipulate the variable's memory directly.

In Conclusion

Despite some efficiency pitfalls and potential for misspellings, the Datafier is a powerful tool. It allows on-the-fly creation of slots for your data. And this data is immediately available globally, without any need for recompile of other modules. Datafier also provides a way to make your "global" data thread safe.

It's definitely changing the way I do things.

Download the source

Jeff Slarve is an independent software developer and the creator of the critically-acclaimed In Back automated file safeguard utility. Jeff has been a Clarion developer since 1991, and is a member of the group formerly known as Team TopSpeed.

Reader Comments

Posted on Thursday, February 05, 2009 by Geoff Robinson

Jeff this is absolutely BRILLIANT.

Some years back I bemoaned the lack of associative arrays in Clarion - see my article "Determining Gender With Clarion" at

http://www.clarionmag.com/cmag/v4/v4n11gender1.html

I can't wait to play with this!

thanks

Geoff R

Add a comment

Clarion Magazine

Using Vista's Complete PC Backup

by Jane Fleming

Published 2009-01-29

"8-Bit". That's what my 27-year old co-worker calls me. Of course, he started with computers at age 6 and thinks nothing of his main SQL database being 13 gigs (plus log files). I think I may have been psychologically maimed writing assembly code for microcontrollers with 1K of EEPROM and 112 bytes of RAM (including the stack). No wonder I'm a wreck. I wonder whom I can sue....

Which is a bit of a ramble to introduce my primitive backup practices. I'm basically an XCOPY girl. My various data folders get backed up to USB drives or network shares, but not the entire PC. I have a copy of Ghost 2000 on my shelf, but never really trusted it.

So one morning about a year ago when I looked across my desk and saw an ominous text screen on my Sager laptop that tersely proclaimed "OPERATING SYSTEM NOT FOUND", with razor-like insight I emoted, "this is not good." When I got home, I popped the drive out of the laptop and connected it to one of those extremely handy USB-to-SATA/IDE drive cables plugged into another machine. The drive had had three logical drives. Now, drives D: and E: were readable with the USB cable, but drive C: showed as "unknown".

I've found Sager to be quite fast shipping replacement parts, and as they're also in California I had a new hard drive the next day. But then, of course, came the rebuild. Reinstall XP. Reinstall Office. Reinstall dozens of other programs. Find license key information for said dozens of other programs. Go through product activation for said dozens. Suck down updates, patches, and service packs.

The good news was that I wound up with no loss of data. The bad news was that it was close to a week before I had everything restored and reconfigured to the point that the laptop seemed "comfortable" again.

Last summer, I convinced myself that I could justify buying a new laptop to replace the repaired one, which was now more than 3 years old. I bought a Sager with 1920x1200 screen, dual-core Pentium, 4GB RAM, and no operating system. (No O/S is not an option on their website - you have to ask for it. They took $97 off the price.) I've wound up with several Vista and XP licenses, and have a Technet subscription, so I figured I'd configure the machine the way I wanted.

I installed 64-bit Vista Ultimate. I opted for VMWare to run 32-bit XP for the Clarion IDE. VMWare supports USB devices in guest machines, which was one factor that sold me on it versus Microsoft's virtualization solution. I gave the virtual machine 16 gigs of disk space - 8 each for the C: and D: drives.

And life was good. But... this VM stuff can be addicting. It's nice to be able to take snapshots and try various configurations. But that all takes disk space on the host.

At the time I'd bought the laptop, 200GB was the largest 7200RPM drive they offered. Now a 320GB, 7200RPM Hitachi was calling my name. But how would I get from Here to There?

Vista Ultimate has a few features missing in some of the other versions. BitLocker - Microsoft's whole-disk encryption system - is one (it's also available in Vista Enterprise, to which only large-volume customers generally have access). Unfortunately, the Sager doesn't include a Trusted Platform Module 1.2 chip, so I'd need to use a flash drive for the decryption key every time I booted. I wasn't sure I wanted to do that, but I set up my partitions so BitLocker would be something I could decide to implement later. (Bitlocker requires a separate 1.5GB primary partition that's marked Active).

Vista Ultimate (like Business and Enterprise) also has a feature called "Complete PC Backup". Microsoft is taking images seriously. If you've actually installed Vista from scratch, you may have observed that it installs a good bit quicker than older versions of Windows. Instead of copying across a gazillion files from the i386 folder on the installation media, Microsoft is now using imaging technology. A "WIM" (Windows Imaging Format) image is copied across onto the target hard drive.

Complete PC Backup uses another imaging technology, which Microsoft also uses in their Virtual Server products. This technology saves the backup as a VHD (Virtual Hard Drive) file. The backup can be initiated from within Vista, and a "bare metal" restore can be initiated from the installation DVD.

Complete PC Backup is happiest backing up onto another "local" hard drive. It can back up to a network share, but there are some limitations (see the additional reading at the end). I've got a 500 GB Iomega external USB drive, and decided to use that.

The backup needs to be restored to a machine of similar architecture (it's not wise to try to restore a 64-bit O/S onto a 32-bit machine), and the new hard drive(s) needs to be at least as large as the one(s) that were backed up. As I was installing a larger drive on the same machine, obviously neither of those constraints was an issue.

Complete PC Backup can be started from the GUI, but ol' 8-Bit opted to use the command-line. I decided first to take a look at how Vista saw my existing disk drive. Diskpart showed the first partition (my BitLocker partition, which Windows calls my D: drive) as 1.5GB and the remainder of the disk as what is the C: drive (Figure 1). NOTE: any of the command-line screen shots in blue were started with a "Run as administrator" right-click. Not knowing whether it might interfere with an image-based backup system, I decided to turn off Diskeeper's automatic background defragmentation temporarily.

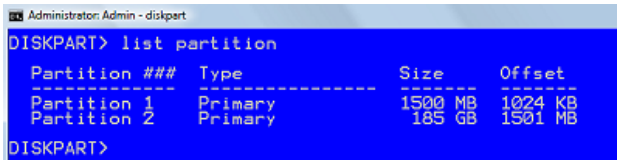


Figure 1. Diskpart on my original hard drive.

So a bit after 4 PM on a Sunday afternoon, I typed the magic mantra

wbadmin start backup -backuptarget:h: -allcritical -quiet

to start the Windows backup admin program (wbadmin) and sent it into motion creating my backup (Figure 2).

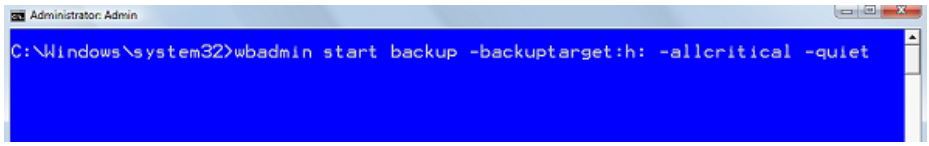


Figure 2. Command to start backup.

The Iomega drive is drive H: on my machine. One doesn't specify a folder with wbadmin, just the drive letter. The -allcritical command is supposed to tell the backup program to back up all the partitions that Windows requires. In that my "Bitlocker" partition isn't actually being used for BitLocker, I was curious how that would be treated.

That question was quickly answered, as the D: drive (partition 1) was backed up first (Figure 3). Not a great surprise, as Windows Disk Manager does show the D: drive as my System partition (see Figure 16).

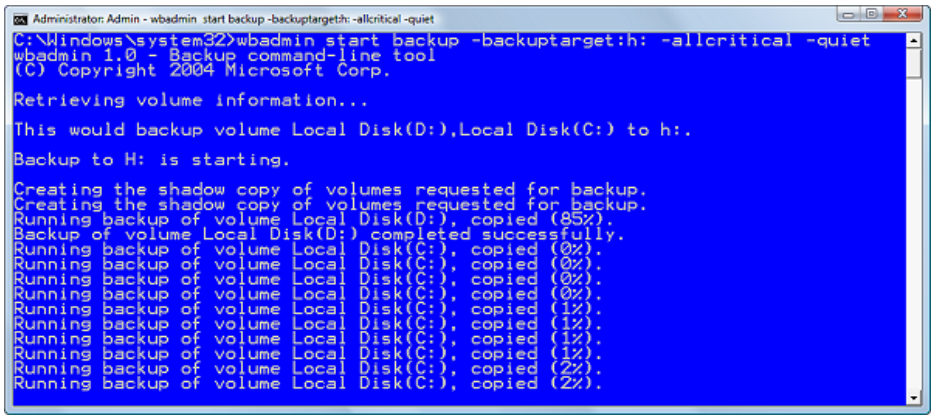


Figure 3. Backup running. D: is complete, C: has begun.

The backup took about two hours, at the end of which wbadmin announced its success (Figure 4).

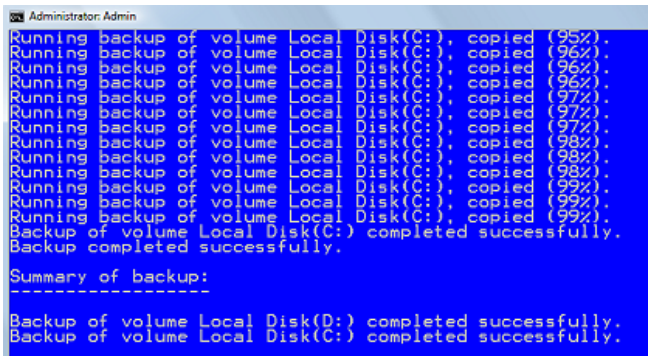


Figure 4. Backup is finished.

There had been a good bit less than 200GB free on my USB drive, but the backup fit without a problem. The roughly 130GB in use on the hard drive had been compressed to about 95GB. Once you've backed up a drive, you can make subsequent backups that are differential - in other words, only changes since the previous backup are recorded. The drive image(s) are stored in a folder that wbadmin creates called WindowsImageBackup. And within that is a folder named for the computer being backed up (my Vista laptop is ingeniously named VL1). By default, Windows sets permissions so as to keep everybody out (Figure 5).

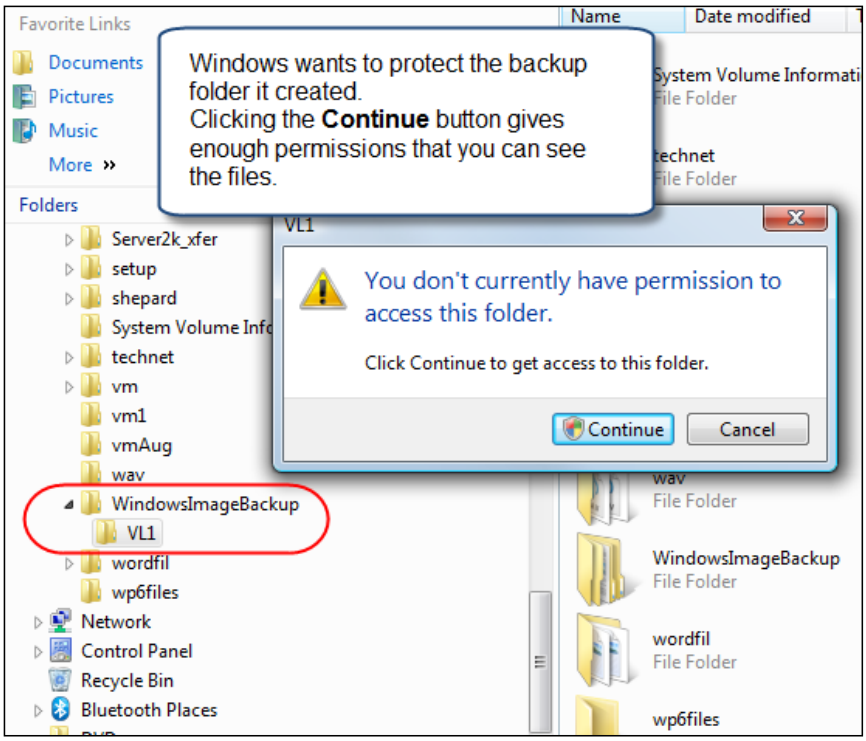


Figure 5. The backup folder(s) on the USB drive.

After taking ownership you can see the files, although you may still want to give yourself Modify permissions if you plan to mount the backup drive (as explained later). Figure 6 is a shot not of my original backup, but after I've backed up the laptop on two more occasions. The two virtual hard disk (VHD) files are a little bigger than 100GB and contain all three backups.

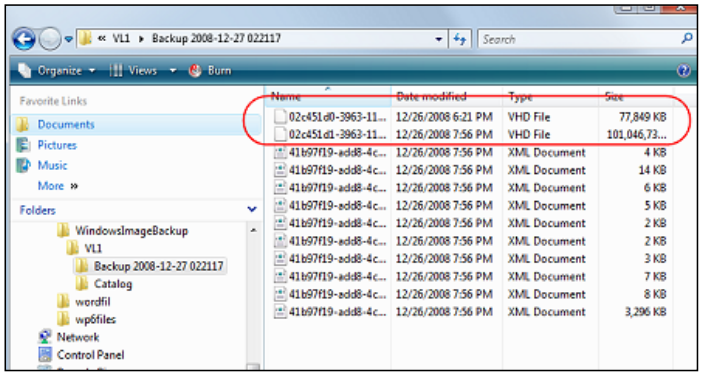


Figure 6. The backup data files on the USB drive.

This article, of course, is about the hard disk swap process, for which I only did the original backup. But as a quick aside - the wbadmin tool can also be used to show you what your backup contains. Here's its output showing the three backups I now have and their status (Figure 7). Again, specifying my H: drive, the command is:

wbadmin get versions -backuptarget:h:

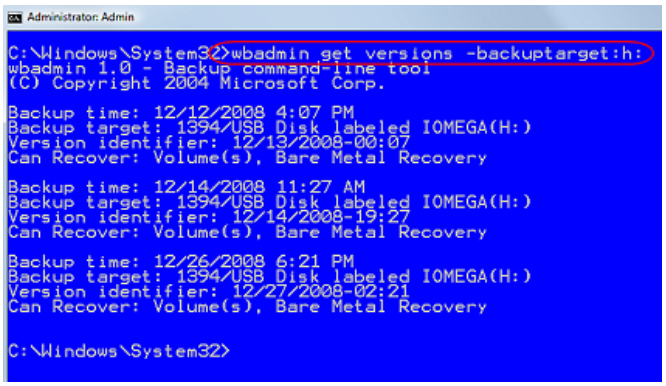


Figure 7. Showing 3 backups available for Bare Metal restore.

The **version identifier** is something you'll need when using the command-line to restore a backup. Note that the version identifiers use Universal Time rather than local time. But I'll be showing how I used the GUI to restore the backup to my new hard drive, so I don't need to know the version identifiers.

Having backed up my drive, it was time to power down the laptop, swap in the new drive, and boot the Vista DVD. After clicking through the initial language/keyboard selection screen, the Install Now screen appeared (Figure 8). In that I wasn't doing a regular installation, I selected Repair your computer.

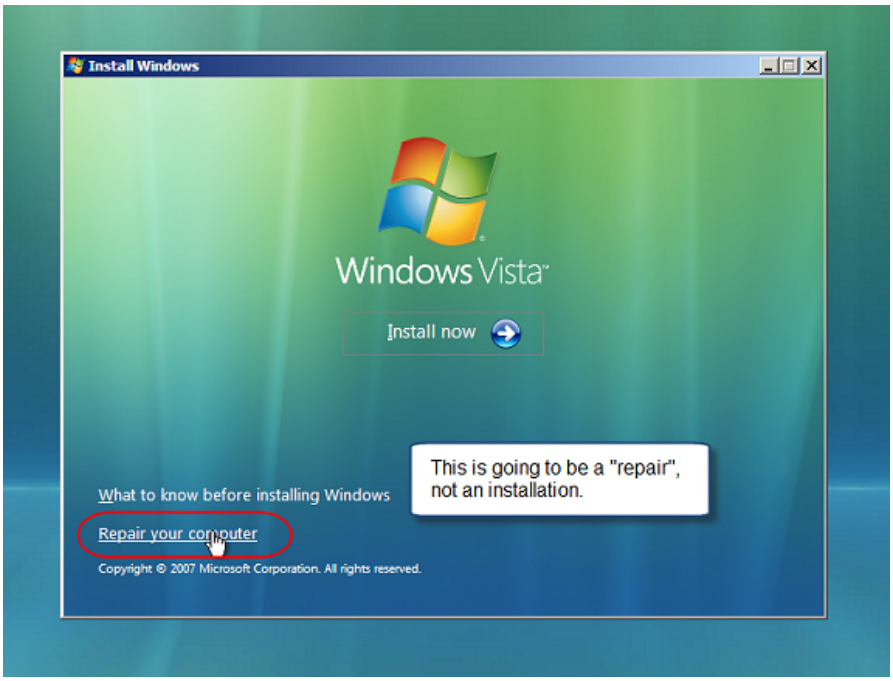


Figure 8. Select Repair, not Install.

As this is not actually a "repair", the next screen that would let me choose which installation to repair was empty because the hard drive was blank (Figure 9). I clicked Next.

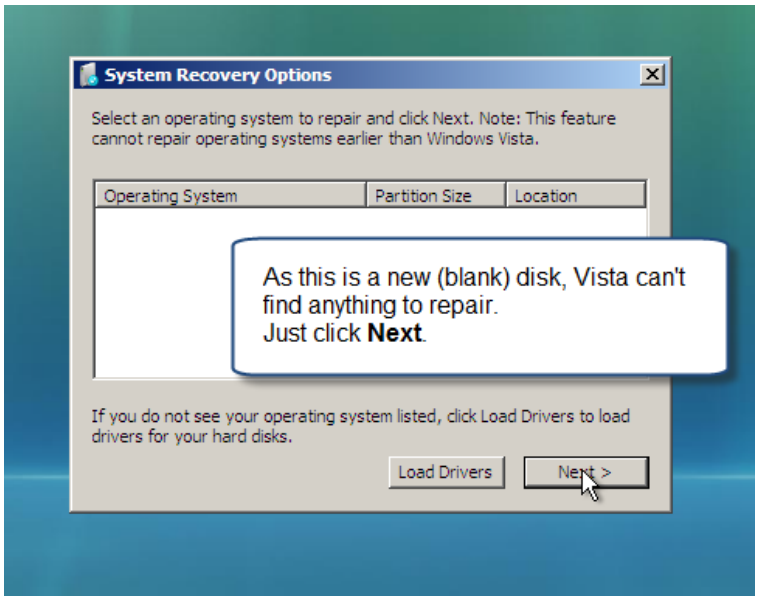


Figure 9. No existing installation, so just click Next.

That brought up a screen with various repair tools (Figure 10). Microsoft is finally including a RAM checker in its installation program!

If I were restoring from a network share, I'd need to use the command prompt to start the network and type startnet. After that, I could use the command line or the GUI to navigate to the backup. But since my backup was on a local (USB) drive, I clicked the Complete PC Restore option.

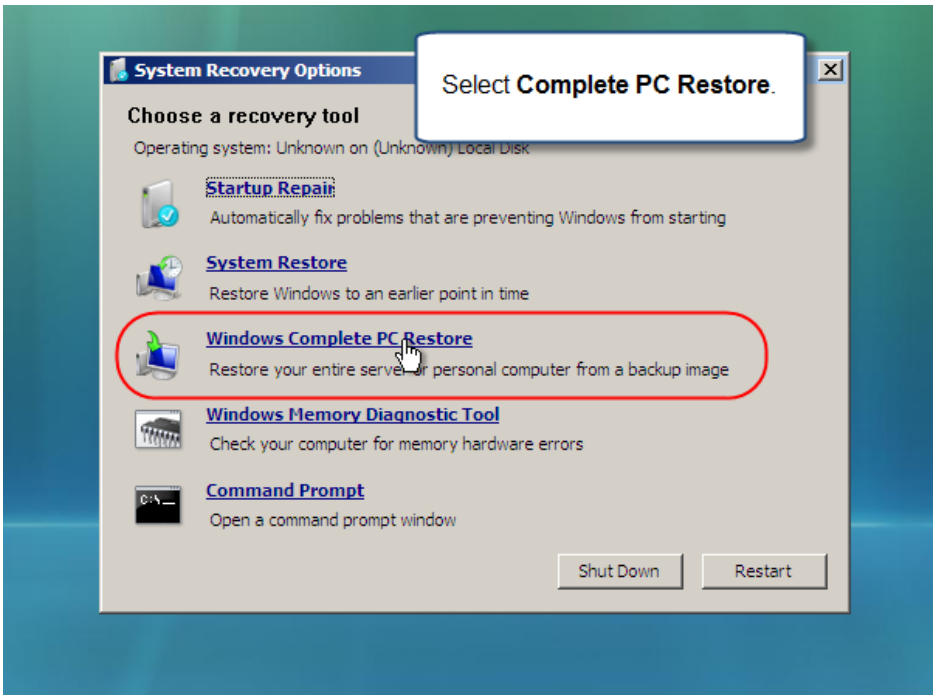


Figure 10. Select Complete PC Restore.

The GUI defaults to the most recent backup it finds (Figure 11). If that isn't the one you want, click the Restore a different backup radio button which will let you select other local backups or type in a UNC network path.

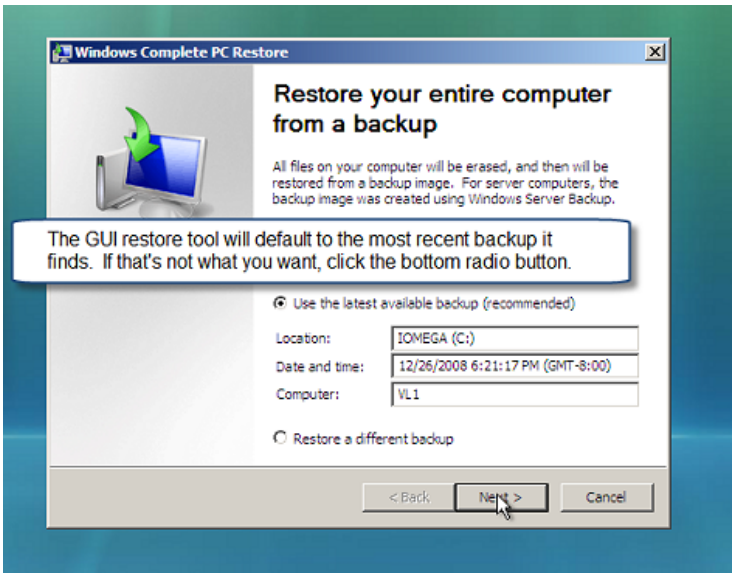


Figure 11. Verify that this is what you want to restore.

Now began the "are you sure" warnings (Figure 12).

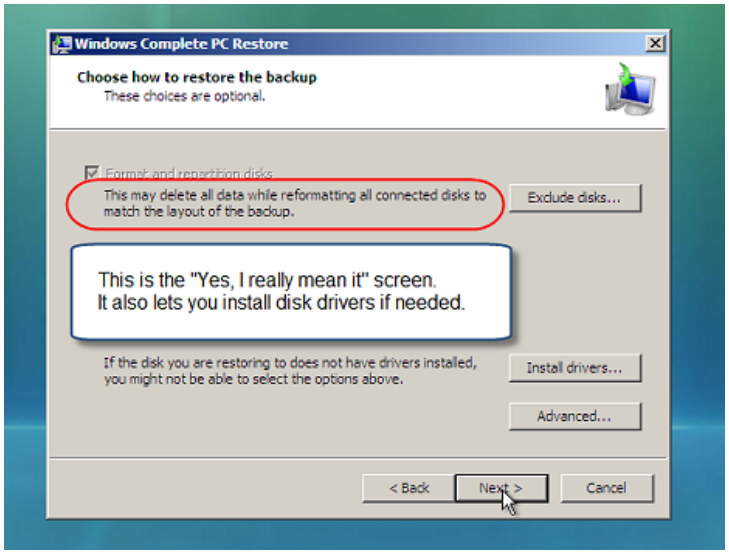


Figure 12. Comfirm the restore.

On Vista, you no longer click the Start button to shut down the machine. So I guess it makes sense in the interests of carrying on a tradition of incongruity that you click the Finish button to *begin* the restore... (Figure 13).

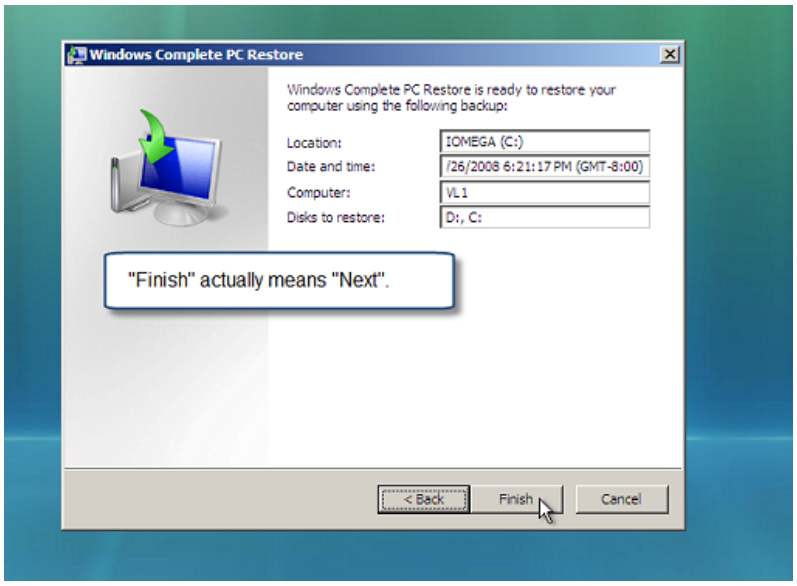


Figure 13. Ready to begin the restore.

Well, better reassure the wizard one more time (Figure 14).

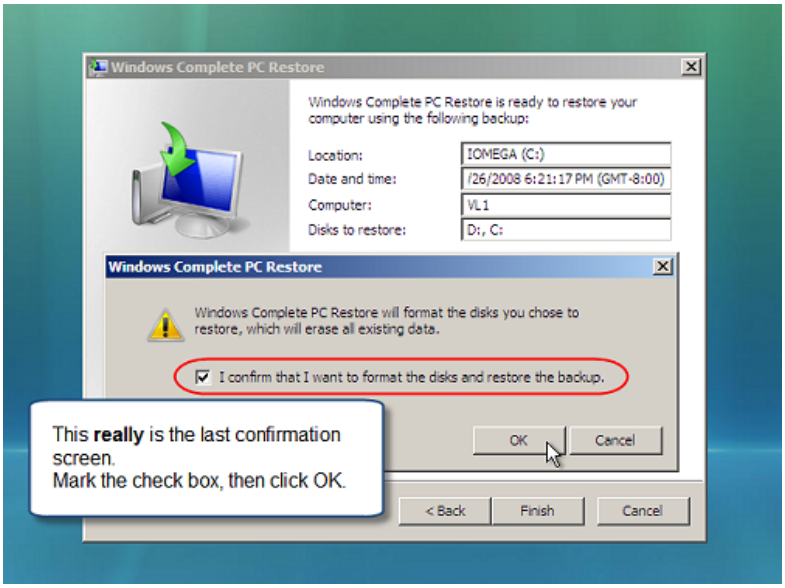


Figure 14. Yes, I really, really, really mean it!

And Complete PC Restore was off and running.

It seemed time to pour myself a glass of wine. Which I did.

I didn't keep close watch of the time, but wandered back into the room a while later (about 7:30) and saw a Vista logon screen displayed. And I actually stood there and giggled. Way cool and virtually painless! The restore had evidently taken less time than the backup. I'd begun the backup at about 4, and now had my machine up and running with its new hard drive.

So what about my "extra" BitLocker partition? Diskpart showed it had been correctly restored as partition 1 (Figure 15).

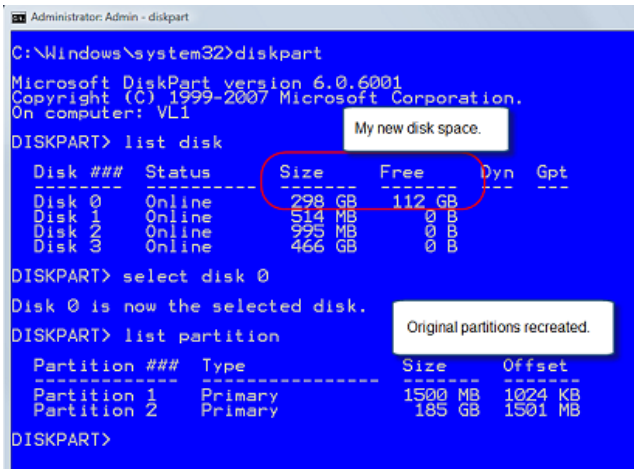


Figure 15. 112GB unallocated, and my original partitions recreated.

I haven't yet decided whether to extend my C: drive (Figure 16), or to leave it at its present size and create a new logical drive in the additional space. The latter might be a better strategy, now that I've broken myself of the habit of relying totally on XCOPY for my backups. I could dedicate that to drive images for my VMWare virtual machines and do separate XCOPY backups of those on an as-needed basis and not have those VM files be part of the bare-metal-restorable images. (When they do away with XCOPY, it really will be time for me to retire!)

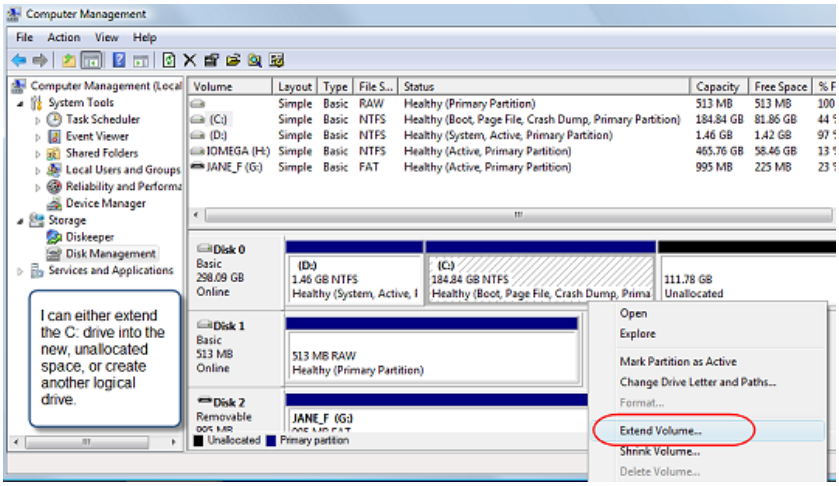


Figure 16. I can extend C: or create a new logical drive (view full size image**)**

For the wimps and girly-men reading, yes, you CAN initiate the backup from the GUI (Figure 17). And once you've done so, you can also use the wbadmin commands (get status, stop job, etc.) If you get tired of watching the output of get status, whether the backup was initiated from the GUI or from the command-line, Ctrl+C will stop the display but the backup process will continue. You can invoke wbadmin get status to resume showing the details of the backup that's underway.

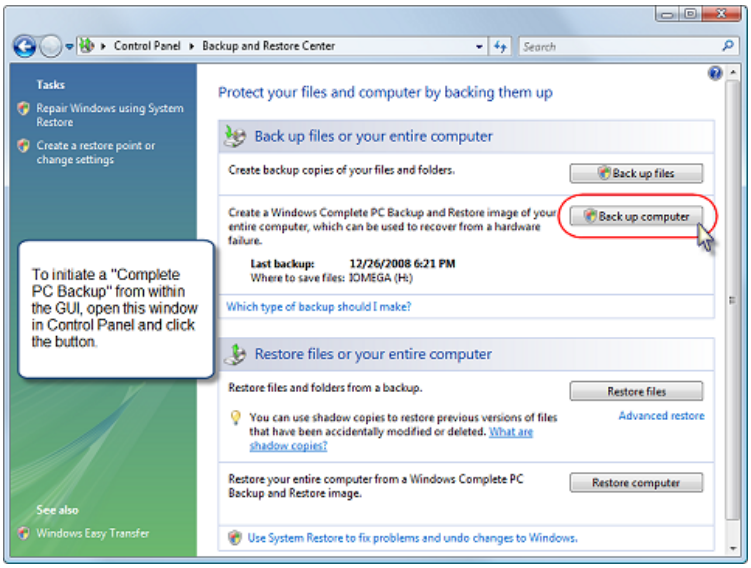


Figure 17. Starting backup from Control Panel.

I just want to touch on one more aspect of the Complete PC Backup thing. As I mentioned earlier, the backup is stored in VHD format. As with VMWare virtual disks, this disk can also be mounted so that you can use Windows Explorer (or the command prompt) to retrieve older files.

To mount the virtual drive, you need to download Microsoft's (free!) virtual server. Search the Microsoft download site for Virtual Server 2005 R2 SP1. Be sure you download the correct version for your hardware (32-bit or 64-bit).

You're not going to install the whole virtual server, though - just the tool that lets you mount a VHD. After you click through the first few screens (the product ID code is preinstalled), select Custom (Figure 18).

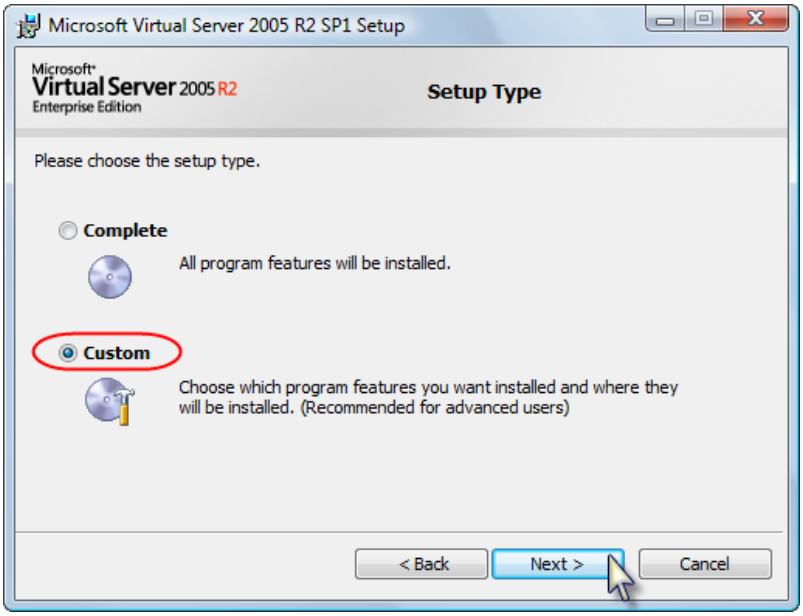


Figure 18. Select Custom installation.

Just choose the VHD Mount tool (Figure 19).

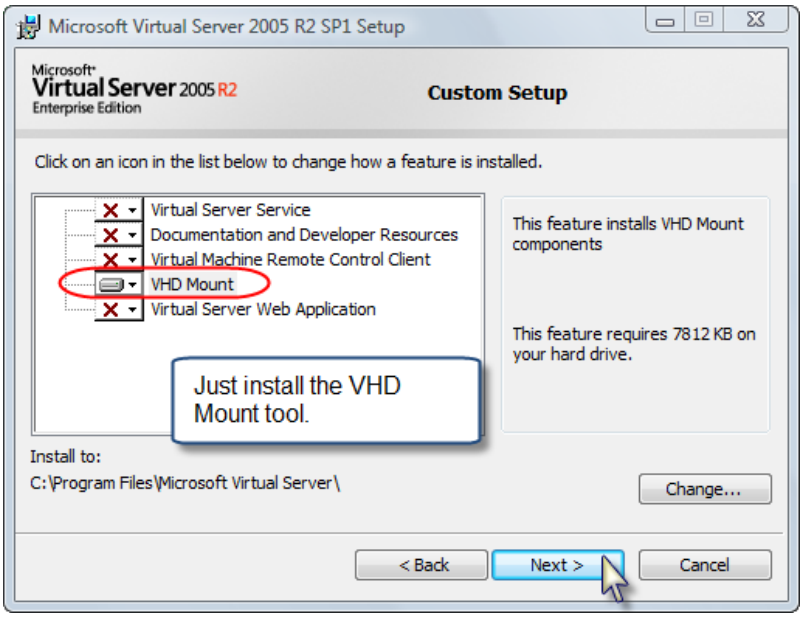


Figure 19. Select VHD Mount.

Unfortunately, all did not go smoothly for me. I don't know if it's a 64-bit Vista thing, but I was unable to use the tool to mount a drive. A little web searching turned up instructions for installing the VHD device driver manually (see the references at the end of this article). Basically, after the installation thinks it's completed you still need to do two things:

1. Use the Add Hardware wizard in Control Panel. Browse to the folder where VHDMount was installed and select the vhdbus.inf file (Figure 20) and go through confirming and clicking OK.
2. Then, try to mount a virtual disk. When it fails, open Device Manager, click the errant item (showing a yellow exclamation point in Figure 21), opt to reinstall the driver, then browse to the same INF file

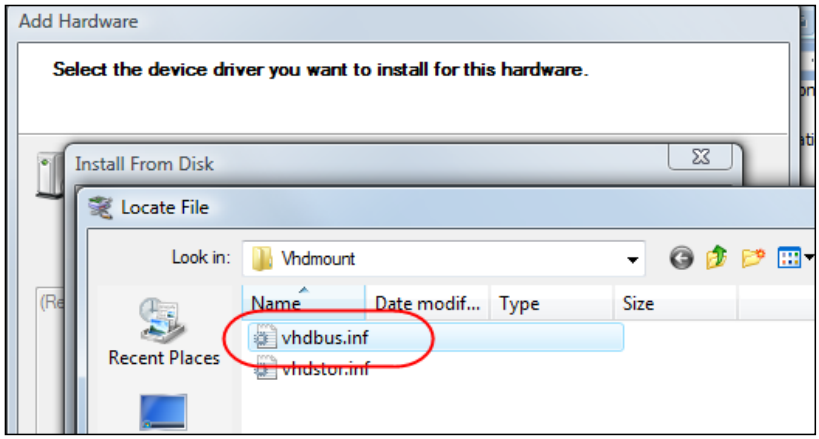


Figure 20. Selecting INF file in Add Hardware wizard.

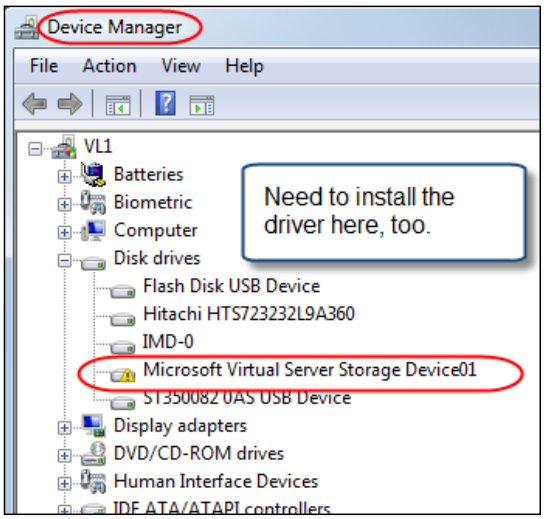


Figure 21. Reinstall driver in Device Manager also.

Once the driver has finally been installed enough times, you can mount your backup virtual hard drive(s). Looking back at Figure 6, you see that the two VHD files are within the same folder and have weird GUID-looking names. The name of the backup of the D: drive begins with 02c451d0 and the backup of C: begins with 02c451d1 I took ownership (Figure 5), but you may also need to give yourself appropriate permissions to the file(s).

As the folder where I installed VHDMount is not in my path, I opened an elevated command prompt in that folder. The command-line parameters are a bit ugly, but Vista's auto-complete makes it less onerous. The /m switch says to mount this VHD.

My backup is on my H: drive, so I started with the syntax

```
vhdmount /m h:\w
```

and then hit the Tab key. Autocomplete filled in the command with the first item (file or folder) it found that begans with "w". As my first folder on that USB drive beginning with "w" is "wav", I hit Tab a second time and it filleds in the second folder, which is the correct one (Figure 22).

Figure 22. The beginning of the command to mount the virtual drive.

I typed a backslash (****) after the command and hit Tab again, and Autocomplete supplied the name of the folder containing my backups (VL1). Another backslash, then Tab. Getting close - I was in the Backup 2008-12-27-022117 folder (look back at Figure 6). Then I typed a backslash, and Tab again. The first time I hit Tab autocomplete presented me with the D: drive backup (the file name whose first segment ends with a zero). I hit Tab again and the command specified the correct backup file for the C: drive I was trying to mount (Figure 23).

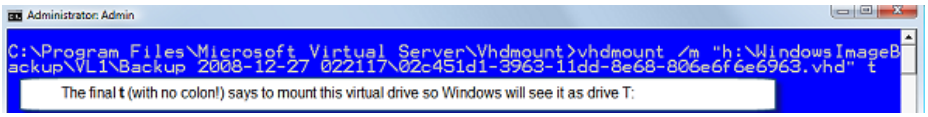


Figure 23. The command is complete, and ready for me to hit Enter.

I decided to mount this VHD as drive T: on my machine, hence the final letter t in the command. Note that if you type a colon after the t the command will fail. With the full command complete, I hit Enter. A few seconds later, success (Figure 24).

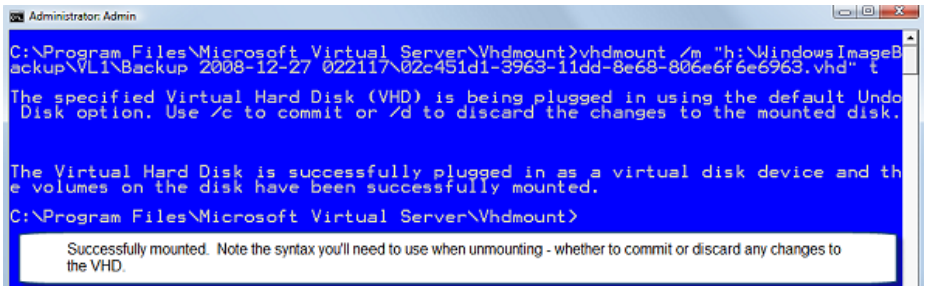


Figure 24. Virtual drive successfully mounted.

Now the mounted drive appeared in Windows as my T: drive (Figure 25).

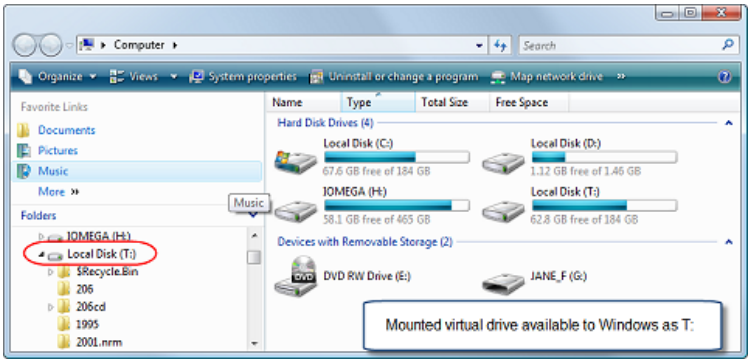


Figure 25. Mounted virtual drive appearing as T: to Windows (view full size image**)**

When I was done with T: I typed a similar command to unmount the drive (Figure 26). The **/u** switch says to unmount the VHD, and the /d switch says to discard any changes to the VHD. F3 is an old DOS trick to avoid retyping commands (and the up-arrow key is a similar but more powerful shortcut). As my command window was still open, I hit F3, then changed /m to /u and replaced my drive letter with /d.

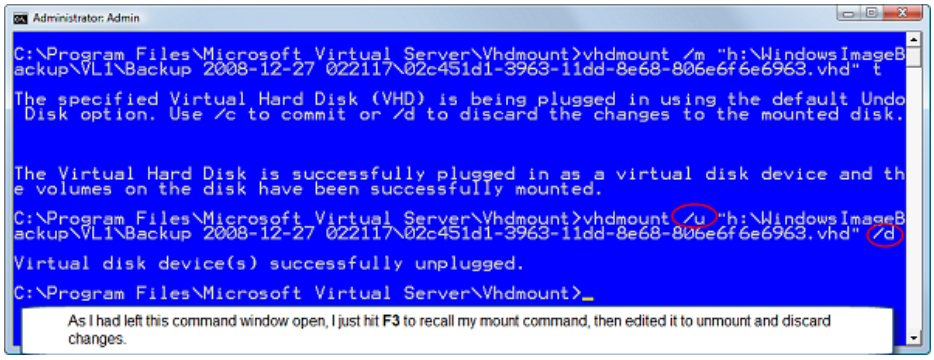


Figure 26. Unmounting the virtual drive once I'm finished with it.

Summary

Yes, there are a number of other disk imaging tools available. But if you're using Vista Ultimate or Business, I'd suggest you look into the one you've already paid for. Complete PC Backup has a couple of advantages, including being able to rebuild a PC in one step from the installation CD and being able to mount and get files from a backup image file.

For more information check out the references below. And now if you'll excuse me, I'm off to back up my disk again.

Additional reading

- Technet Magazine article on Vista Backup and Restore technologies.
- Hotfix for VHDmount.
- Another VHDmount article (from a user perspective).
- More details on installing and using VHDmount.
- Mark Minasi's newsletters are a great resource (require a free login to read). Two pertinent issues are:
    - Backing up to a network share.
    - Restoring from a network share.

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

Clarion Magazine

Managing And Debugging Builds With External Project Files

by Rick Martin

Published 2009-01-23

As you may already know, you can add an external PRJ or PR project file to an APP file's project settings (see Figure 1). The settings specified in the external project are included in the project settings for the APP. One of the things I use this for is controlling whether or not Debug or Release mode is turned on for the APP.

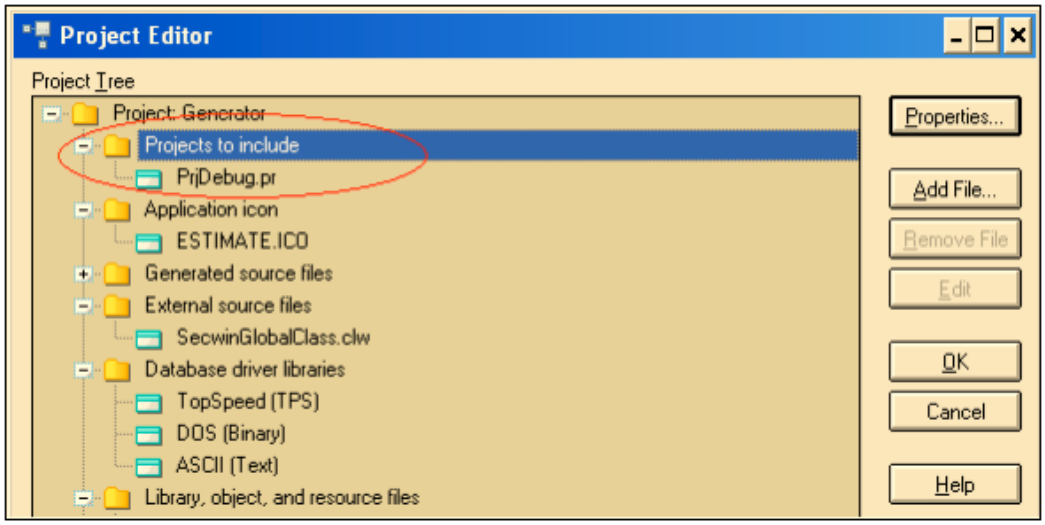


Figure 1. An included project

I work on systems where the number of APP files involved ranges from 29 to 200. Ensuring debug is turned off for all these APP files when building a release version is time consuming, tedious and prone to error. As a member of the Lazy Programmers' Society there is nothing I hate more than having to remember to do tasks that are unrelated to what I am trying to accomplish, or having to do repetitive tasks. It is bad enough when I have to remember for myself, but working with a team of developers can be even more difficult. It is hard for team members to remember to switch an APP to release mode before checking it in. Their focus is on the bug fix or enhancement.

When I am developing I like my APP files to have debug turned on for two reasons. One, I like the debugger. Even though the debugger has gotten a bad reputation over the years I find it very useful and it is far more stable in Clarion 6 than in prior versions. Two, I use a lot of ASSERT statements. Asserts let you test assumptions in your code and display a message if the test condition is false. The message is only displayed when debug is enabled for the APP.

One place where assert statements are particularly useful is with Prop:SQL statements. When you use a string to build a SQL statement the compiler does not process it for typos, incorrect labels/tables or syntax errors. These problems will not be caught until runtime when the string is passed via Prop:Sql to the database engine. Often, if you have an error in your SQL statement you'll simply get no records or sometimes all records, especially if you don't check for errorcode(). For example, look at the complexity of the statement below.

```
DummyTable{Prop:Sql} = |
  'Select Sum(IsNull(BRT.BidUnitPrice,0)*IsNull(QAD.Quantity,0)) '&|
  'From QuantityAdjustment QAD '&|
  ' Join BidResults BRT '&|
  'On BRT.BidItemUnique = QAD.BidItemUnique AND BRT.ProjectBiddersUnique = |
  '&AwardedProjectBiddersUnique&' '&|
  ' Join ProjectBidders PBR '&|
  'On BRT.ProjectBiddersUnique = PBR.PBRUnique '&|
  'Where PBR.ProdjectUnique = '&CTE:ProjectUnique&' '&|
  'And dbo.shGetDate(QAD.AdjDateTime) <= <39>'&Format(CTE:TestDate,@D10-)&|
  '<39> And QAD.Approved = 1'&|
  'And QAD.BidItemUnique In ( Select CTD.BidItemUnique '&|
    'From ContingencyTest CTE Join ContingencyTestDetail CTD '&|
    'ON CTD.ContingencyTestUnique = CTE.CTEUnique '&|
      'Where CTE.CTEUnique = '&CTE:CTEUnique&' And CTD.Quantity <> 0)'
ASSERT(~ERRORCODE(),CLIP(ERROR()) & ' - ' & FILEERROR())
```

It is easy to make a mistake in all that code. When the statement is assigned to Prop:SQL the statement is passed to the backend database and the execution plan is created. If there are any problems the RTL will set ErrorCode(). Error() and FileError() will have the information you need to determine what's wrong with your statement. In fact, I like Assert statements so much that I have a keyboard macro setup to automatically insert the Assert statement in the code above. Figure 2 shows an SQL error message displayed by Assert.

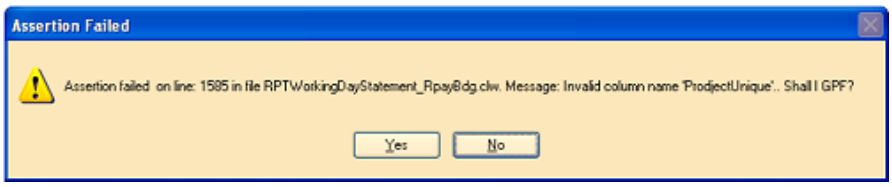


Figure 2. An SQL error message shown via Assert

While the message in Figure 2 is great for me as the developer it certainly isn't something I ever want my end-users seeing, which leads me back to using an external project in all my APP files. I add PrjDebug.PR to every APP file. On my development system it turns debug on for every APP. On the official build system it turns debug off and release on.

PrjDebug.PR is just a text file with commands for the project system.

Here are the contents on the development system:

```
-- Set this pragma to full or off (case sensitive) to force APPs to
-- compile in debug or not
-- Comment out the pragma lines to allow the setting in the App to take effect
#pragma debug(vid=>full)
--#set RELEASE = on
--#pragma debug(vid=>off)
```

On the build workstation it looks like this:

```
-- Set this pragma to full or off (case sensitive) to force APPs to
-- compile in debug or not
-- Comment out the pragma lines to allow the setting in the App to take effect
-- #pragma debug(vid=>full)
#set RELEASE = on
#pragma debug(vid=>off)
```

All lines beginning with -- are comment lines.

With these files in place the programs will always be compiled in release mode on the build system without me every having to think about it again.

If you don't have a build server you can still use this approach. First, create two versions of the project file, one for debug, the other for release. Then copy the appropriate version over top of PrjDebug.PR as needed. If you use a separate copy of PrjDebug.PR for each app, you'll probably want to use a batch file for this task, but you can also place PrjDebug.PR in a standard location and have each app's project include the same file.

There are a lot of other options you can add to the project file. You can find the complete list and a description of the project file definition in AdvancedTopics&ReferenceGuide.PDF in the Docs folder.

For example you can add a project define:

```
#pragma define(_AppIsDefined_=>1)
```

or link in a resource:

```
#pragma link("MyIcon.ico")
```

I hope this article gives you some food for thought on how you can use external projects to automate and control your development environment.

Rick Martin has been programming systems and applications for over 20 years. Outside of work Rick avoids high-tech as much as possible and enjoys hobbies in golf, woodworking, and the outdoors. He and his wife Cathi are enjoying their new grandson who lives with his mom just minutes away. Rick makes his home in beautiful Chico, California.

Reader Comments

Posted on Monday, January 26, 2009 by Nardus Swanevelder

Very Cool, I was looking for an easy solution to turn debug on and off.

Thanks Rick

Posted on Wednesday, January 28, 2009 by Jeffrey Slarve

Very cool.

Add a comment

Clarion Magazine

Using Multiple Clarion Versions In C7

by Steven Parker

Published 2009-01-23

I think it was at the last Devcon, during a discussion of the features of Clarion 7, that it was announced that Clarion 7 would be able to build applications in any version of Clarion. The .APP file would not be backwardly compatible, of course, but the resulting DLLs and EXEs would be built using the templates and binaries of the selected version.

Of course, the desired version of Clarion would have to be installed on the development PC. No, Clarion 7 was not going to include all those previous versions of Clarion.

I have it on good authority that, while significant, this was not thought of as a blockbuster feature. But the audience was very much impressed by the idea. Recent response to the first AppGen release shows that this feature is indeed *very* much valued by Clarioneers. To my thinking, this makes Clarion 7 not simply a new version of Clarion but an all inclusive shell for all my Clarion work.

Yes, it was mentioned that this single IDE would eventually wrap both Clarion 7 and Clarion.NET. To this, every version of Clarion I have installed is added. Clarion 7, in large measure, becomes a wrapper around any of a number Clarion variants (even different builds of a given release).

This means that I can maintain applications in any flavor of Clarion I want to with, as was *de rigueur* in the old days, but a single entry point. But I don't have to open and switch between multiple instances of Clarion (though, it turns out, I can open multiple instances of Clarion 7 without corrupting my template registry).

Setting Clarion Targets

After installing the first Clarion 7 AppGen release, I selected Tools | Options | Clarion | Versions from the main menu and, lo and behold, all of the versions of Clarion that I have installed on this machine appeared (Figure 1).

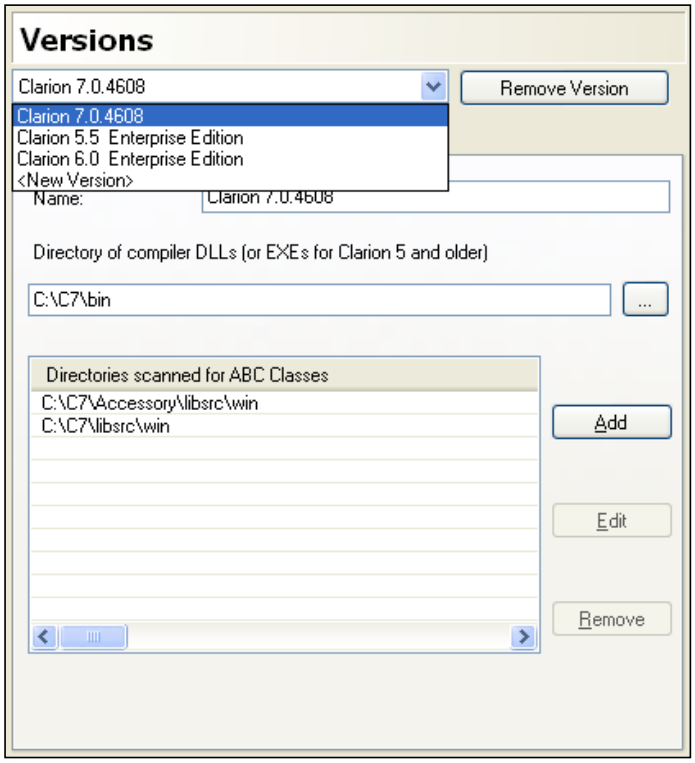


Figure 1. Registration of other Clarion versions

(Note: You can also manually add Clarion versions in the event C7 doesn't find them, or you install them after you install C7.)

I thought I was done. I thought I could just open an application from a previous version of Clarion and build it in that version. Not so. When I opened a 6.3 app, the IDE showed it as 7 and built accordingly. (Clarion 7, for DLL mode apps, copies the required Clarion DLLs into the output directory. This makes it very easy to tell what happened: just look at the Clarion DLLs copied in.)

My next thought was that, when opening the app and allowing it to convert, first I needed select the target version from Tools | Options | Clarion | Versions. Again, not so.

Building into previous versions of Clarion is not quite so simple.

In the first place, the target version is not set from Tools | Options | Clarion | Versions. The target version is set from Build | Set Clarion Version, as in Figure 2.

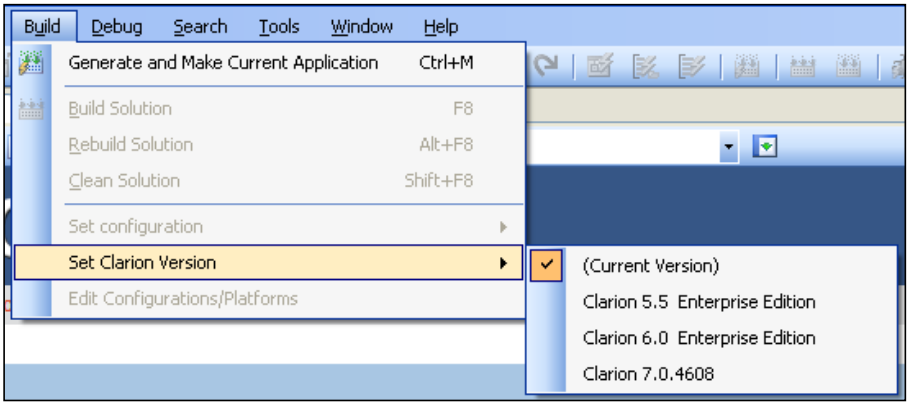


Figure 2. Setting the target version

But the sequence begins even before you set the target version from the Build menu.

First close any open applications and any open solutions. Yes, those are two separate steps.

Next set the target version from the Build menu.

Then open the .APP. (Notice that I didn't say open the solution (.sln) or the project (.cwproj), but the APP file - I'm specifically talking about how to bring a C6 .APP into C7 and stick with the C6 templates, libsrc, DLLs, etc. Later on I'll talk about switching versions once the app is converted.)

If I close the app, when I return to it, the selected version of Clarion "sticks" (Figure 3) That is, if I check the selected version from the Build menu or the C7 title bar, while in the app, it will be the version I selected before opening it (the target version cannot be changed while the .APP is opened). Even if I open an app tied to a different version in between. That's because the Clarion version used for that app is stored inside a solution preferences file in the %AppData% \SoftVelocity\Clarion\7.0\preferences directory (which raises other issues about sharing apps; as C7 is still in beta the location of this information may yet change).

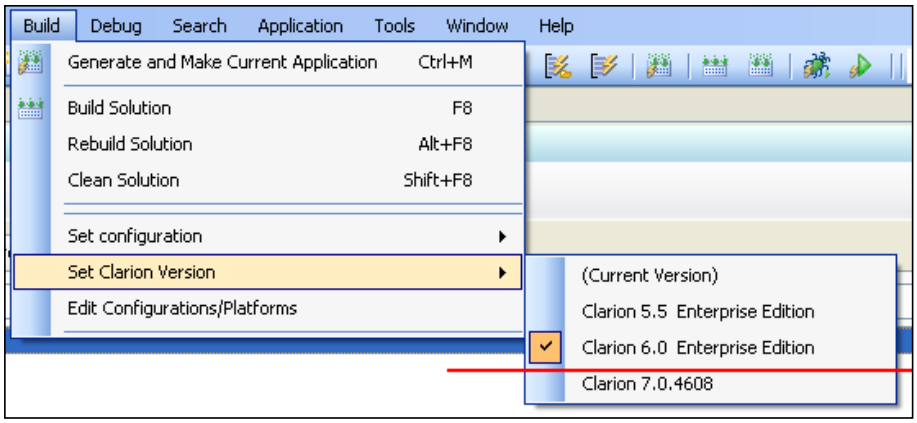


Figure 3: Reopening C6 mode application

(This is what the Current Version option in the Build menu obviously means. Therefore, to upgrade to a new version of Clarion, click on the desired, newer, version - instead of Current Version -- and open the .APP.)

Templates

With the information about how to set the Clarion version in hand, I set my version to Clarion 5.5 and tried to open an app from a previous Clarion Magazine article. The next prompt (Figure 4) was not precisely what I expected.

Figure 4. Opening a Clarion 5.5 .APP

It seems that, while Clarion 7 recognizes that I have previous versions of Clarion installed, Clarion 7 doesn't register the templates for those previous versions.

So, with the version set to a previous version (from the Build menu), I chose Tools | Edit Template Registry and press the Register button. Clarion 7 *does* read the redirection file of the target version, so it goes to the correct template directory.

I started with my Clarion 6 (9056) install. All templates in my \Clarion6\Template directory registered correctly. The same for my \Clarion6\3rdParty\template template directory; all registered correctly.

I was not so fortunate with my 5.5 templates. When I tried to register them, several failed (Figure 5).

Figure 5. Registering ABCHAIN.TPL for 5.5

Figure 6 shows the template errors.

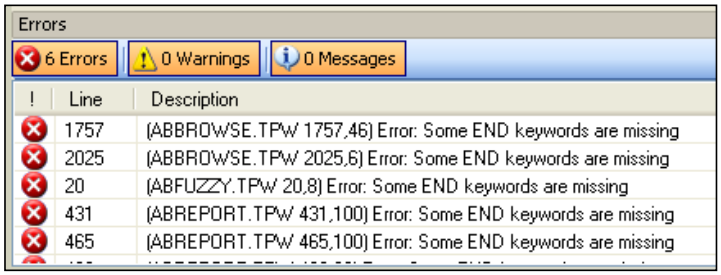


Figure 6. Registration errors

There is, as I recall, mention of this in one of the documents accompanying the installation. It specifically states that the C7 template parser is much more strict than earlier versions, especially about missing END statements. Well, it is.

When I click on one of the lines in the error list, I am taken to a .TPW file where, indeed, and END statement is expected:

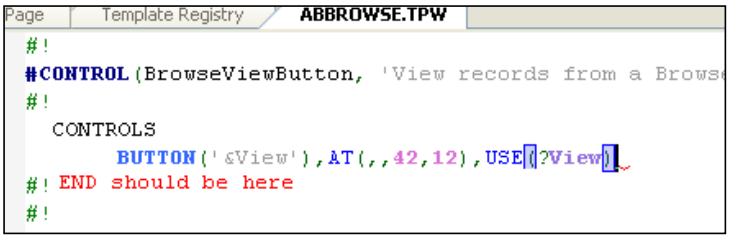


Figure 7. Missing END

I did not finish this exercise for 5.5. I'm saving that for a demonstration at our next users group meeting. But, it ought to work.

ABC classes

Clicking Tools | Options | Clarion | Versions, I dropped the list box to see the details Clarion 7 stored. I noted a list titled Directories scanned for ABC Classes (Figure 7).

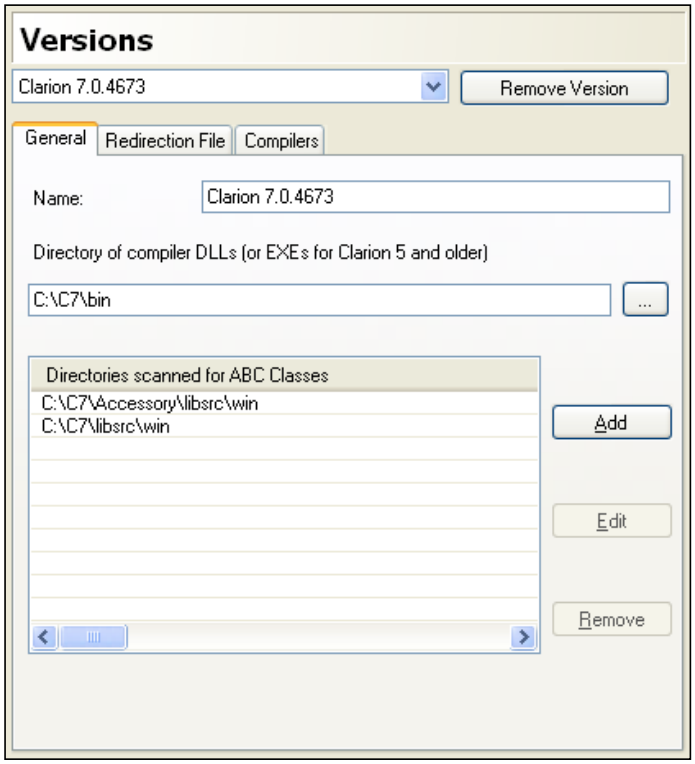


Figure 7. ABC class directories

This list includes both the standard libsrc directory and the Accessories (third party) libsrc directory for Clarion 7 (I do have several third party templates installed already). But for 6 and 5.5, it only includes the standard libsrc directory.

A newsgroup posting by SuRF states that the third party libsrc entry is not necessary for previous versions. Clarion 7 will use the version's .RED file and, if that file is correctly configured, there ought be no problems. However, SuRF notes that making the entry does give you more flexibility.

Pressing the Add button, on the right does allow me to navigate, via a Windows' select dialog, to the directory I want to select. But, I cannot save the directory as my selection. A bug report has been submitted.

There are two ways to get the directory into the directory search list. One is to highlight any file in that directory and press the Open button. Only the directory name will be included in the search list entry.

The other is to edit the configuration file directly. It turns out that the settings, here, are in \Documents and Settings\<userName>\Application Data\SoftVelocity\Clarion\7.0*ClarionProperties.xml*. You can also get to this by pasting %appdata%\SoftVelocity\Clarion\7.0\ into Windows Explorer (or via Ctrl-Esc | Run, or on Vista, just via Ctrl-Esc).

After opening ClarionProperties.xml in NotePad, locate a node called Properties name="Clarion.Versions." Part of my copy of this file reads:

```
<Properties name="Clarion.Versions">
  <Properties name="Clarion 5.5  Enterprise Edition">
    ...
    <libsrc value="C:\C55\libsrc" />
    ...
  </Properties>
  <Properties name="Clarion 6.0  Enterprise Edition">
    ...
    <libsrc value="C:\CLARION6\libsrc" />
```

...

etc.

Edit the libsrc value entry, changing

```
<libsrc value="C:\C55\libsrc" />
```

to

```
<libsrc value="C:\C55\libsrc;C:\C55\3rdParty\libsrc" />
```

And change

```
<libsrc value="C:\CLARION6\libsrc" />
```

to

```
<libsrc value="C:\Clarion6\3rdParty\libsrc;C:\CLARION6\libsrc" />
```

Return to the Versions scanned list and both the standard and third party libsrc directories are correctly listed.

Why? I'm thinking that this would work on a PC where I "simply" copy my old Clarion directories, without benefit of "installation" or where I have installed from the CD, thus updating WIN.INI and allowing Clarion 7 to "see" the version, but not creating a good .RED file. (Given the ability to add versions, mentioned above, this may be less a worry than I'm thinking it is now. But, "better safe &.")

A caveat

Dave Harms has noted that when you set the Clarion version for a solution, that information is stored in a related XML file in the %AppData%\Roaming\SoftVelocity\Clarion\7.0\preferences directory. If you close C7, delete the preferences file, and restart C7, when you next load the application it will default to the current version of C7. This also implies that simply giving an application to another developer will not preserve the version information; you'll need to tell that person which Clarion version to use.

Summary

The steps to use previously installed versions of Clarion, then, are:

1. Set the build target using the Build | Set Version menu
2. Register templates from the Tools menu
3. Add missing END statements
4. Add \3rdParty\libsrc to Directories scanned for ABC Classes

Does it work? Yes, it does. When I opened and built GTL, a Clarion 6 app, in Clarion 7 but as a Clarion 6 app, the resulting EXE was exactly the same size as the EXE produced by Clarion 6. Switching it to DLL mode, Clarion C60 DLLs ended up in the output directory.

Cool.

Steve Parker started his professional life as a Philosopher but now tries to imitate a Clarion developer. He has been

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

Reader Comments

Add a comment

Clarion Magazine

C7 AppGen CSP Beta Report

by Dave Harms

Published 2009-01-12

SoftVelocity has released a second CSP AppGen beta, and as I haven't yet had a chance to report on either of the CSP releases I'll cover both in this article, in no particular order.

As I mentioned in the ClarionMag blog, there are two beta newsgroups, one for the third party folks and one for the CSP participants. There have been over a thousand messages posted in the beta newsgroups since the initial Christmas release. In general those postings fall into the following categories

- Education - learning how things work/where things are in the new IDE.
- Usability - some tasks are faster in C7, others aren't (yet). Achieving a C6-like level of keyboard shortcuts is a hot topic.
- Installation problems - leftovers from previous betas can cause some perplexing issues.
- Architectural issues - there are very few of these, but one that comes up often is multi-app solutions and the time to load the registry for each.
- Bugs - okay, I forgot this category in the blog post.

There have also been a few nuggets of information revealed which I'll pass along a little later in this article.

Education

The new IDE has a lot of new functionality, and some of the old functionality has a different interface. As a result, some of the bugs being reported aren't bugs at all. For instance, someone posted a complaint about converting applications from C6 to C7, where the applications should stay in C6 mode (that is, although the internal app format is upgraded to C7, the IDE uses the C6 templates, compiler, libraries etc.). I figured I'd better check that out. So I applied one of my own templates to a C6 app, set my C7 IDE to C6 mode, and opened the app. C7 tried converting the C6 app but couldn't find the template. Aha, a bug, I thought - C7 is converting it with the C7 templates.

In fact the problem was with the wetware, not the software; I'd neglected to register the template in the special template registry C7 uses for my C6 templates. Once I did that, I was able to convert the C6 application just fine, using the C6 templates. In fact, Steve Parker submitted an article on this subject while I was on my Christmas holiday, and if I'd read it I wouldn't have made that mistake, as Steve was quick to point out. (Look for Steve's article later this month.) Keep in mind, however, that as mentioned in previous articles the new template parser is stricter than the C6 and earlier parsers, so most templates will need a little tweaking. Check with your template vendor(s) for updates.

Speaking of Steve Parker, the good doctor spent some time going through the Learning Clarion lessons and reported a few issues. According to Bob Foreman the Learning Clarion lessons are the best way to get used to the new IDE, and Steve endorses that point of view. But as Steve found out you can expect a few wrinkles during the beta as the IDE described by the lessons is still a bit of a moving target.

There's new functionality, but sometimes old functionality seems to be missing. For instance, there are no buttons in the current release to reorder local data items. You can, however, drag/drop to reorder queue fields in local data. If I recall correctly the move up/down buttons will be added as well (and certainly should be), but this is a classic example of functionality that appears missing but is merely different.

And sometimes a bug report has nothing to do with C7 at all. One tester reported a problem with a "too many segdef" error which he had hoped would go away in C7. As it turns out this was a legacy (Clarion) template chain application. Bob Z explained that this isn't a compiler/linker issue per se, but something that can be addressed in the templates. In fact, the ABC templates already solve the problem:

> The change to the generated code was very straightforward, we just generate a series of modules containing nothing but the file definitions, limiting each module to a preset number of FILE definitions. Using this approach there is theoretically no upper limit to the number of FILEs your program may contain.

There were a couple of requests for a way to specify build order in a multi-project solution. In fact this is largely unnecessary, since the build system (based on MSBuild) looks for dependencies and begins by building projects that have no dependencies and then works its way up.

If you have circular references then you can simply do the build several times. And that prompted a discussion about the safety of circular references. At first I agreed with one of the other posters that a reference is a reference, and it really shouldn't matter if it's circular. But then Russ Eggen reminded me of an article I published only a little over a year ago in which George Lehman removed all circular references from his apps and thereby eliminated all "memory could not be read" GPFs. Two other developers have reported similar experiences. George had this to add:

> My understanding at the time, based on a quote from RZ, was that the order of deallocation of resources is indeterminate when you have circular references. If DLL A references a procedure in DLL B, and DLL B references a different procedure in DLL A, regardless of which DLL is unloaded first, the other may try to free resources via a kill method or something similar for a procedure or some other element from the first that is no longer there.
>
> I can personally testify that eliminating circular references dropped my field GPFs by 90% and has indeed completely eliminated the 'memory could not be read' GPFs.

So there you have it. The build system is your friend; if it finds circular references, get rid of them. It's not an easy task, but George's article will show you the way (George also references an earlier article by Steffen Rasmussen.

Quite a few developers have puzzled over what's happened to the C6 Global Properties. They're still there, but if you go to the Global Properties tab for an application you'll see what in C6 are the Application Options.

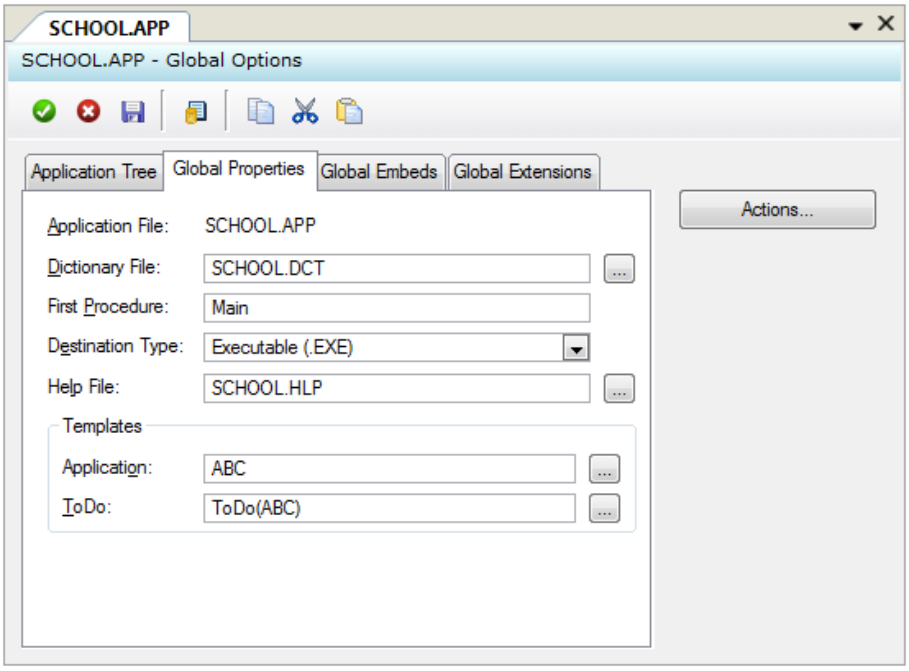


Figure 1. Locating the global properties

To get to the familiar C6 Global Properties settings you need to click on the Actions... button.

Usability

For some of us there's still too much mouse clicking required in the new IDE, and sometimes a series of clicks can be downright inconvenient. If, say, you're in the window designer, the mouse way out is to click on the green arrow a couple of times. But click once too many times and you close the application.

In both CSP betas when you have an app open and the focus is on the application pad the main menu becomes unresponsive; that is, it doesn't close when you click away from it, and you can't use the keyboard shortcuts (not that there are all that many of these anyway). For folks who rely on the keyboard there's still a ways to go.

There have been a couple of complaints about the IDE not remembering the last directory. Yes, you can specify the default application directory but I'm glad to see others are as annoyed as I am with this one. It's been difficult to convince the dev team that this is an option worth having, but I have hope they'll come around. It's not something that in the end consumes a lot of time, but as I've said in the newsgroups it's often the little things like this that make an IDE a real joy to use.

The IDE still needs tidying up. Menu items aren't always consistent with the names of the windows they open; there are some visual bugs related to resizing, and so on.

One tester wanted to know why code completion is available in the embeditor but not when you go to a single embed point. Z's response:

> When you invoke the Embed editor either by the context menu or the toolbar button the complete source code for the Procedure is generated on the spot, and then you are presented with all the embed points with a context (which makes it much easier to see where to place your code). Without the context of complete code for the Procedure there cannot be code completion, except for the list of builtin functions from the RTL because they are static. When you enter an embed from the Embed tree dialog, and you choose a Source type embed you are in an editor session that has no context.

As a long-time user of the embeditor I can't imagine restricting myself to the single-embed approach, but some developers prefer it because it presents less information: you only have to deal with that one embed point, not a bunch of code that may or may not be generated. Some else (I think it was Bob Foreman) pointed out that when you're at the list of embeds (Figure 2) you can choose an embed point and click on Source. That will take you to the desired embed point in the embeditor, which seems like a reasonable compromise.

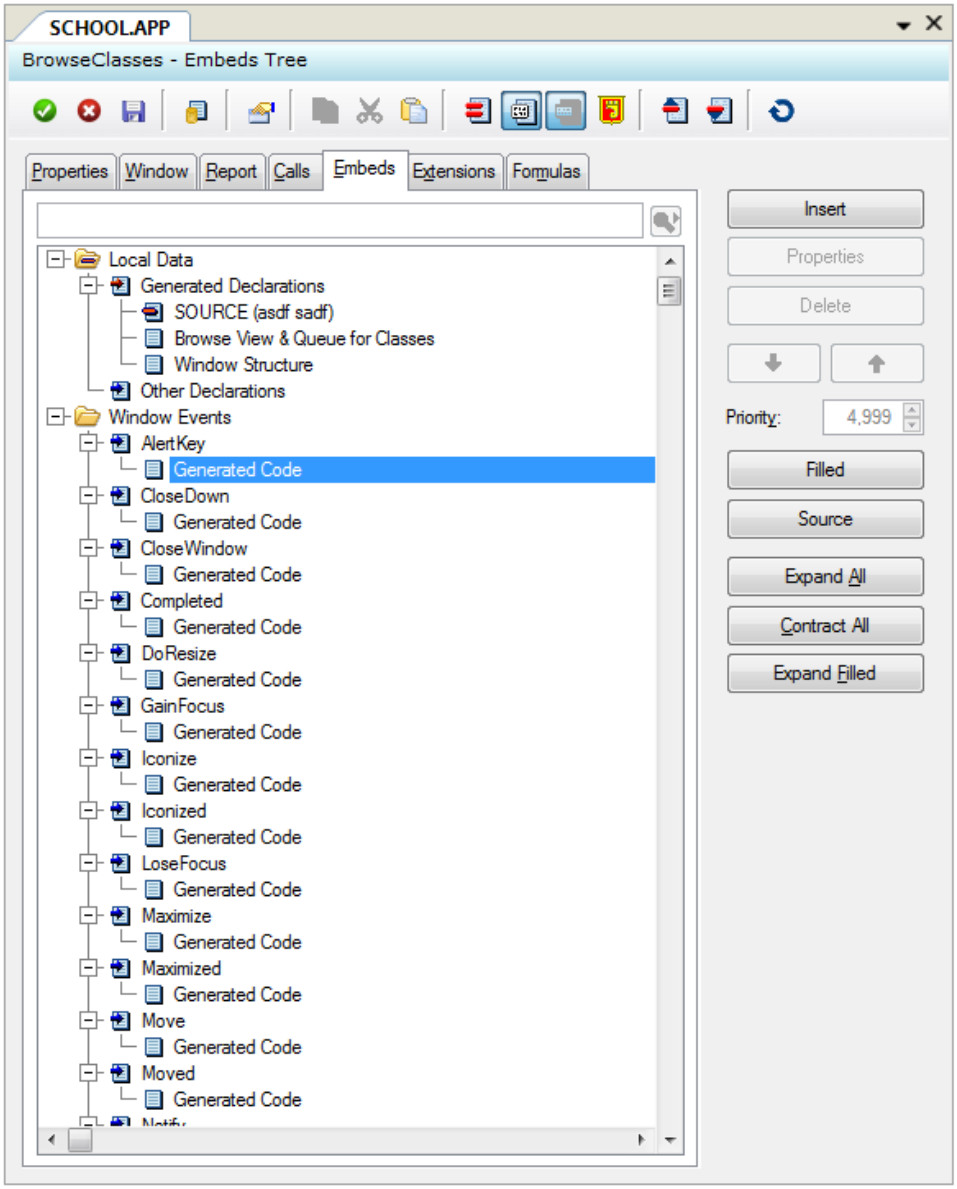


Figure 2. The Source button and the embed list

Architectural issues

Although C7 is much faster than C6 at generating and compiling, the ability to have multi-app solutions has raised some concerns over speed and memory usage. Some developers want to know why, if all the apps in a solution use the same templates and classes, each application has to go through the process of loading up that information. Can't the data be shared?

That's really a design issue, as Bob Z explains:

> While it may seem from the surface a trivial change, unfortunately it is not. Every APP gets its own unique instance of the Generator running on a distinct thread, and each APP has its own unique symbol table. And the support for 'versions' other then the one running is an added complication. So we're not making any promises that this will change before the gold release. Though I think we may be able optimize the loading before the gold release.

As an aside, it's important to note that even if your master solution has a hundred apps in it you can create much

smaller solutions with only those apps you really need to work on. When you first convert an application the IDE will attempt to convert any related applications which can be a time-consuming process, but after that you can put apps in whatever solutions you like (think of a many to many relationship) without the IDE including anything for you. But you will need to manually add in whatever libs or related apps are necessary for your solution to compile.

Installation problems

Several installation problems have been reported. You may have problems running the IDE if you choose a custom installation directory like c:\Clarion7, and you could run into other issues if the IDE has leftover data from a previous C7 build. Bob Z showed a nifty way to get to the directory containing the IDE's data. Instead of trying to drill all the way down to the appropriate folder, just use the system variable together with the SoftVelocity subdir:

```
%APPDATA%\SoftVelocity
```

You can enter this via Ctrl-Esc | Run (or on Vista, just Ctrl-Esc), or you can type it into Windows Explorer. There are other symbols you can use:

- %ALLUSERSPROFILE%
- %PROGRAMFILES%
- %SYSTEMROOT%
- %USERPROFILE%
- %WINDIR%

Although I'm showing these symbols in all caps, they aren't case sensitive.

Bug reports

In the first CSP beta there was some confusion over editing errors. In that release any time you got an error in embedded code and you double-clicked on that error, the IDE would take you to the generated source. There's a setting under Tools | Application Options to force this behavior, but in the first CSP beta you were taken to the generated source automatically. In the second beta you get an unhandled exception (UHE) if you double-click on an error that's in embedded code. So it looks like work is being done on this capability; perhaps embeditor error handling will be ready for the next beta.

Some users have reported problems deleting procedures, resulting in app corruption. There are still some known problems with LIKE and key components.

As I mentioned earlier, double-clicking on compile errors doesn't yet take you to the embeditor. I think there's a good chance we'll have that for the next beta, and it is the one major item mentioned in the release notes as not working yet.

There are certainly more bugs than I've mentioned here, but these are some that have come to the fore in the last two betas.

I'm not seeing a lot of mention of folks posting PTSS reports. That doesn't mean they aren't doing so, but clearly it's best to add a PTSS entry, preferably with an example that clearly demonstrates the bug. I do see some folks posting about bugs but who are unwilling to add a PTSS report. I *really* don't understand this attitude.

Odds and ends

TXA formats have changed in C7. The TXA version is now 32, whereas my 9055 build of C6 issues a version 28 TXA. DEPEND statements (which are a sort of counter) have a value that's often one less than in the earlier TXA. There are some extra or expanded WHEN statements, and some text is formatted differently which makes comparison difficult (WINDOWs, particularly).

The most pervasive TXA change, however, is that IDENTs, which uniquely identify controls, have been replaced by GUIDs (globally unique identifiers).

Although I haven't tried yet, it seems pretty clear that a TXA from C7 will not import into C6. That shouldn't be a big problem since you can use older versions of the Clarion templates, source and compiler/linker with C7.

And somehow I'd never noticed that C7 doesn't export/import TXDs. The new format is XML. Here's a snippet (with line breaks added)

```
<?xml version="1.0" encoding="UTF-8"?>
<Dictionary Name="SCHOOL" Version="1">
  <DictionaryVersion Version="1" Description="Initial version"/>
  <Table Guid="{2a3dd7ca-b2a3-45bd-ac26-e087b8d719ad}" Ident="1" ↵
   Name="Students" Prefix="STU" Driver="TOPSPEED" ↵
   Create="true" Thread="true">
    <Audit CreateUser="Administrator" CreateDate=" 8 OCT 1998" ↵
     CreateTime="11:30:43AM" CreateVersionNumber="1" ↵
     ModifiedUser="Administrator" ModifiedDate="30 SEP 2003" ↵
     ModifiedTime=" 4:04:14PM" ModifiedVersionNumber="1"/>
    <Field Guid="{3480ca90-8673-4ecb-8e20-bbcde4a35ec9}" ↵
     Ident="3" Name="Number" DataType="LONG" Size="4" ↵
     ScreenPicture="@P###-##-####P" ScreenPrompt="&amp;Number:" ↵
     ReportHeading="Number" Justification="RIGHT" Offset="1">
      <Audit CreateUser="Administrator" CreateDate=" 8 OCT 1998" ↵
       CreateTime="11:30:43AM" CreateVersionNumber="1" ↵
       ModifiedUser="Administrator" ModifiedVersionNumber="1"/>
      <Validity Check="NONZERO"/>
      <WindowControl ControlText="PROMPT(&apos;&amp;Number:&apos;)↵
       ,USE(?STU:Number:Prompt)"/>
      <WindowControl ControlText="ENTRY(@P###-##-####P)↵
       ,USE(STU:Number),RIGHT(1),REQ"/>
      <ReportControl ControlText="STRING(@P###-##-####P)↵
       ,USE(STU:Number),RIGHT(1)"/>
    </Field>
```

Since SV has seen fit to go ahead with XML for dictionary exports (a good move) I'm not sure what the benefit is of still using TXAs as the app export format - perhaps we'll see that changed to XML as well.

Tips

When asked about the new IDE's Component Inspector, Z responded that it was contributed to the IDE by Oakland Software. Unsurprisingly, this is the same component inspector you find the SharpDevelop IDE. You use the Component Inspector to explore ActiveX controls and .Net assemblies. See the product page for more information on what the inspector can do.

If you want to change the location where the EXE is created, it's recommended you do so via a local redirection file instead of setting the output directory in the build options.

I've long wondered how the template language's #PROJECT statement would handle the new XML project files, particular in Clarion# where you can have non-compiled files and dependent files that should show up in the project as linked to another file. Scott Ferrett supplied the answer a few days ago in a newsgroup post:

> The #PROJECT extension is designed to support any MSBuild project file. So it will work if you are creating a Clarion

project, a Clarion.NET project, a C# project or anything else. The first parameter translates to the element type, the name within the brackets translates to the value of the Include element and each x=y pair afterwards translates to a metadata element. Eg to generate this into a .csproj file

```
<Compile Include="Properties\Settings.Designer.cs">
  <AutoGen>True</AutoGen>
  <DesignTimeSharedInput>True</DesignTimeSharedInput>
  <DependentUpon>Settings.settings</DependentUpon>
</Compile>
```

You would do (editor's note: line breaks added):

```
#PROJECT('Compile(Properties\Settings.Designer.cs),AutoGen=True,DesignTimeSharedInput=True,
DependentUpon=Settings.settings')
```

If you want to know what is valid for Clarion, just fiddle with the properties within the IDE and look in the .cwproj file to see what happens.

That should do the job just fine, I think.

Summary

Progress continues to be made on the new AppGen and on the IDE in general. There don't seem to be many template compatibility issues left to iron out, although as I mentioned earlier most templates will need at least some minor massaging before they'll register in C7. That won't be an issue if you're moving to C7 and your vendor supports C7 versions of the templates you use. It's most likely to be a hassle for developers who wish to stay with older releases and are using templates that are no longer maintained.

The "getting started" help file contained a notice about this being a candidate release, but that information doesn't appear anywhere in the IDE itself and this would appear to be a case of premature documentation. There are still some issues to be ironed out, and I would look for an overall stable release or two before the word "candidate" gets mentioned. It's steady progress, and I can't say how long the beta will take, but SV is cranking out the builds with regularity and there have been surprisingly few regressions along the way.

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

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

Tips Vol 5 and upcoming articles

Direct link

Posted Tuesday, January 27, 2009 by Dave Harms

I've had a few questions about when the Tips Vol 5 book is coming out. Normally we do the pre-sales about 4-6 weeks before the expected ship date, which meant the book would have shipped around the middle of January. But in retrospect it probably wasn't a good idea to run this process over Christmas (and the CSP beta releases, as welcome as they were, didn't help either!). But the good news is that production on the book is almost done, and we're hoping to have the first proof order off to the printers around the end of this week, with the final version about a week after that.

There are still three terrific articles coming this week, including a pair by Jane Fleming. Jane revisits the code-signing certificate purchasing process and also brings to light some interesting features in Vista's Complete PC Backup. And Jeff Slarve has a terrific article on the Datafier. What is the Datafier? Think of it as a repository for global data. But it's much more than that. Read the article later this week and see for yourself.

New C7 release

Direct link

Posted Friday, January 23, 2009 by Dave Harms

SoftVelocity has released C7 build 4749 to CSP program participants. This one has a somewhat shorter list of bug fixes, and the newsgroup also seems somewhat quieter than after previous releases (although I haven't quantified this). I take all of this as a positive sign, although I note that the embeditor error handling code still isn't complete. At this point I'm not

seeing anything significantly new about build 4749 (other than that a bunch of stuff has been fixed), and at this point I'm not planning a report.

Well, actually there is this one item in the change document regarding multi-version support:

> FEATURE: The IDE now notifies you when you open a solution using a different Clarion version than is currently selected. You can choose to open in the version the solution was last opened in or the currently selected version.

We're also supposed to get a new Clarion# build next week. That will be welcome, as it's been over three months since the last one. The next Clarion# release will have at least some level of .NET 3.5 support (and keep in mind that 3.5 is 2.0 plus extensions, so you can use Clarion# now with 3.5 provided you're only using the 2.0 classes).

The importance of being earnest... no, I mean accurate

Direct link

Posted Monday, January 19, 2009 by Dave Harms

You might have noticed that the latest survey question changed this morning from "How important to you is the ability to preview a window in the designer?" to "How important to you is the Preview! menu item in the window designer?" This is actually a complete change of survey, not just wording, and any responses to the first survey have been discarded.

While I make no pretense that ClarionMag's surveys are scientific, the first question just seemed a little too imprecise. After all, just having a designer is by definition a way of previewing the window. The real reason I'm asking this question is because the Preview! menu option in the C6 window designer has no counterpart in C7, at least not yet. So I'd like to hear how important Preview! is to Clarion developers.

The month/year ahead

Direct link

Posted Wednesday, January 07, 2009 by Dave Harms

I expect 2009 to be a banner year for Clarion development. Over Christmas, SoftVelocity released the C7 AppGen to CSP program participants, and another update is expected this week. I'm not as sanguine as SV about an imminent gold release, but real progress is being made on the beta.

SoftVelocity has also promised an update to Clarion#, presumably with LINQ support and other goodies. And although I don't have any confirmation from SV, I expect this will be the year we see the first .NET templates.

There have been over a thousand messages posted in the beta newsgroups since the Christmas release. I'll be summarizing the discussion in an article to appear later this week or early next, but in general those postings fall into the following categories

- Education - learning how things work/where things are in the new IDE
- Usability - some tasks are faster in C7, others aren't (yet). Achieving a C6-like level of keyboard shortcuts is a hot topic.
- Installation problems - leftovers from previous betas can cause some perplexing issues
- Architectural issues - there are very few of these, but one that comes up often is multi-app solutions and the time to load the registry for each.

Look for the monthly PDF and source code library updates by Thursday at the latest; the January articles will begin appearing early next week. Some of ClarionMag's most popular authors make appearances this month, covering topics for C6, C7 and .NET.

And be sure to go to the home page and take the annual ClarionMag survey about your business prospects for the new year. I've reviewed the results for previous years, and there's an interesting departure so far (with admittedly only about 30 responses). In any given year around a third of respondents expect their business to stay roughly the same; that's the case this year as well. But in all other years "somewhat better" gets more responses than "much better", and "somewhat worse" gets more responses than "much worse". This time around those response rates have reversed. Given the current economic climate I'm not surprised to see increased pessimism; in any given year 10-15% of respondents expect a decline in their Clarion business. This year that figure is a little over 20%, so far. But those expecting a "much better" year also outnumber those expecting a "somewhat better" year. The extremes are, curiously, more prominent. Perhaps the release of new product from SV has some developers stoked, just as economic issues have others worried. And perhaps the numbers will smooth out after a while and go back to the bell curve.

In any case, recessionary times can actually produce opportunities for software developers. For some companies a slowdown in business means an opportunity to review existing systems and invest in IT, something that's much more difficult and dangerous when everyone's scrambling to keep up with orders. Consulting opportunities may also increase as firms are less willing to take on new full time hires.

Year end price cut trimmed

Direct link

Posted Wednesday, January 07, 2009 by Dave Harms

At the end of last year we dropped our subscription prices by as much as $40. We were able to do this thanks to favorable exchange rates - like many Clarion third party vendors located outside the United States, we price in US dollars for the convenience of both US and worldwide customers.

We'd hope to keep this rate cut on the table a little longer, but exchange rates aren't looking as good right now. As of January 10th we'll be rolling back the year-end discount by a minimum of $10 across the board. That means a one year renewal will cost $119, and a one year new subscription (including all the back issues) will cost $169. Those prices are also subject to change without notice.

Subscribe/renew now
