

Clarion Magazine

Save up to **50% off ebooks.**
Subscription has its rewards.



Clarion News

- o » [SV Blog: Clarion 7.1 Window Previewer](#)
- o » [SV Blog: 32 bit ODBC in the 64-bit Operating System](#)
- o » [Noyantis Support Ticket System](#)
- o » [Noyantis User Forum](#)
- o » [RPM C7 Cosmetic Oversight](#)
- o » [EasyListView 1.02](#)
- o » [ClarionLive! Double Header Week](#)
- o » [Search Engine Profile Exchange November 03 2009 Release](#)
- o » [EasyListView Version 1.02 Coming Soon, New Demo.](#)
- o » [Noyantis Codejock Wrapper Templates Now Codejock v13.2.1 Compatible](#)
- o » [CapeSoft Clarion 7.1 Builds](#)
- o » [Xtreme ToolkitPro, Xtreme SuitePro 13.2.1](#)
- o » [Aussie DevCon Training DVDs](#)
- o » [DockingPane Wrapper Template 1.05](#)
- o » [SkinFramework Wrapper Template 2.00](#)
- o » [ReportControl Wrapper Template 1.11](#)
- o » [TaskPanel Wrapper Template 2.00](#)
- o » [PropertyGrid Wrapper Template 1.10](#)
- o » [CalendarPro Wrapper Template 2.00](#)
- o » [CommandBars Wrapper Template 2.01](#)
- o » [ShortcutBar Wrapper Template 1.21](#)

[\[More news\]](#)

- o » [Clarion.NET FAQ](#)
- o » [Clarion# Language Comparison](#)

[\[More Clarion & .NET\]](#)

Latest Subscriber Content

Source Code Library 2009.10.31 Available

The Clarion Magazine Source Code Library has been updated to include the latest source. Source code subscribers can download the October 2009 update from the [My ClarionMag](#) page. If you're on Vista or Windows 7 please run Lindersoft's [Clarion detection patch](#) first.

Posted Monday, November 09, 2009

Aussie DevCon Training, Part 1

Geoff Robinson reports on the Aussie DevCon training sessions. Topics include new C7.1 IDE options, migrating from C6, and the dictionary editor.

Posted Wednesday, November 11, 2009

Aussie DevCon Training, Part 2

Geoff Robinson reports on the Aussie DevCon training sessions. Topics include wizards and themes, usability improvements, and some browse and form tips and techniques.

Posted Wednesday, November 11, 2009

IP Driver Notes

Last year Dermot Herron wrote an article on IP Driver-enabling existing applications. Since that time he has learned more about the IP driver from several other developers on the Clarion newsgroups.

Posted Monday, November 23, 2009

Guest Editorial: Is The Clarion Paradigm Dead?

Why is the data-centric paradigm important? Because it enables rapid development of business applications. But does this paradigm have a future, and if so, what will that future look like? Robert Barton lays some groundwork for discussion.

Posted Monday, November 23, 2009

The Problem With Embeds, Part 3: The Invoice App

Dave Harms continues his series on how not to write embed code by digging into the Invoice app. The UpdateDetail procedure turns out to be an excellent example of what can go wrong in embedded code.

Posted Thursday, November 26, 2009

Clarion 7.1 In Two Weeks?

A new blog post by SoftVelocity details the fixes and new features in Clarion 7.1, and indicates that a release is expected within two weeks. An update to Clarion# is also expected during that timeframe.

Posted Thursday, November 26, 2009

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

Latest Free Content

- » [Clarion 7.1 In Two Weeks?](#)
- » [ClarionFAQ Update](#)

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

Clarion Sites

Clarion Blogs

ClarionFAQ Update

Here's what's happening at Clarion Magazine's new free-access Frequently Asked Questions site.

Posted Monday, November 30, 2009

Have Your Say On All Things Clarion

Have you ever wanted to have our say about which features should be in Clarion, or what SoftVelocity's priorities ought to be, or have you just wanted to make suggestions that other developers could read and possibly support? Check out [Clarion.UserVoice.com!](#)

Posted Monday, November 30, 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.

[More info](#) • [Subscribe now](#)

Printed Books & E-Books

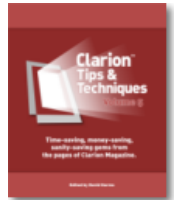
E-Books

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 5 - ISBN 978-0-9784034-1-6](#)
- » [Clarion Tips & Techniques Volume 4 - ISBN 978-0-9784034-0-9](#)
- » [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](#)



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

o » [Clarion Databases & SQL - ISBN: 0-9689553-3-9](#)

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

From The Publisher

About Clarion Magazine

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

Subscriptions

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

Satisfaction Guaranteed

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

Dave Harms

ISSN

Clarion Magazine's ISSN

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

About ISSN

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

Copyright © 1999-2009 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Clarion Magazine

Clarion News

[Search the news archive](#)

DockingPane Wrapper 1.07

Version 1.07 of the DockingPane Wrapper template is available from the members area. Modifications include :Event Handler facility added; Icons now automatically added to Project List (if filename prefixed with a "~"); Icon sizes can now be specified; Tab Caption option added; New methods 'GetPaneClosed, GetPaneEnabled, GetPaneFloating, GetPaneHidden, GetPaneProperty, GetPaneSelected, GetPaneTabCaption, GetPaneTitle, SetPaneIcon, SetPaneProperty, SetPaneSelected, SetPaneTabCaption, SetPaneTabColour, SetPaneTitle, SetPaneTitleTooltip.

Posted Monday, December 07, 2009

Twitter Remote Control With C6/NetTalk

This app, created with Clarion 6/NetTalk, lets you use Twitter, mail or your cell phone as a remote control for your PC.

Posted Monday, December 07, 2009

EasyNaviBar 1.00

EasyNaviBar version 1.00 is now available. EasyNaviBar enables your Clarion applications with the popular MS Outlook 2003/2007 style navigation bar. Six different layout and color styles. Requirements include: Clarion 6.3 / Clarion 7; ABC or Legacy Template Chains; 32-bits only; .NET Framework 3.5 and above; Windows XP and above. Demo available. Price: \$50 (New Subscription: 1 Developer license + 1-year Maintenance Plan).

Posted Monday, December 07, 2009

Ingasoftplus Time Limited 25.00% Discount

Ingasoftplus announces the cutting of prices and a 25% discount on all products (full versions) for the period from 01 Dec till 31 Dec 2009.

Posted Monday, December 07, 2009

CHT Web Client Server Videos

CHT has released two new videos dealing with Web-Client-Server technology, CHT style. These two videos are related to a couple of new demos released with Build 13D1.00. The videos are part of an on-going series discussing web-server/web-client creation with The Clarion Handy Tools.

Posted Monday, December 07, 2009

CHT Build 13D1.00

For an overview of what's new in CHT Build 13D1.00 please see the What's New page.

Posted Monday, December 07, 2009

[SV Blog: Clarion 7.1 Window Previewer](#)

Bob Zaunere shows some screen shots from the new C7 window previewer, to be included in Clarion 7.1. You can open multiple instances of the previewer, and COM objects are live.

Posted Monday, November 23, 2009

[SV Blog: 32 bit ODBC in the 64-bit Operating System](#)

Bob Ricketts posts on how to get access to the 32 bit ODBC administrator when running a 64 bit version of Windows.

Posted Monday, November 23, 2009

[Noyantis Support Ticket System](#)

The new Noyantis Support Ticket System is now online, although there may be further updates to the graphics and layout.

Posted Monday, November 23, 2009

[Noyantis User Forum](#)

The new Noyantis User Forum is now online, although there may be further updates to the graphics and layout.

Posted Monday, November 23, 2009

[RPM C7 Cosmetic Oversight](#)

There is a slight cosmetic oversight when using the latest RPM C7 and C6 in C7, XT templates. It does not affect the use of the templates but it does make the overlapped checkbox a bit hard to read! View the web page for a correction or download the appropriate zip.

Posted Monday, November 23, 2009

[EasyListView 1.02](#)

EasyListView Version 1.02 is now available. Demo app changes include: File explorer example; Export to Excel (TPS example); Export to xml (TPS example); Header Word Wrap (TPS example); Hover selection (File explorer example); Hot tracking (File explorer example); Translucent selection (File explorer example); Translucent hot item (File explorer example); Explorer theme (File explorer example); Flag renderer (File explorer example, Attributes column); Item tooltips (File explorer example); Column header font/color (TPS example); Hot item overlay (TPS example); Check boxes (File explorer example). Template changes include: Header word wrap; FlagRenderer; Use explorer theme; Use translucent hot item; Printing, added description for Header and Footer; Hover selection; Hot tracking; Show header in all views; Show item tooltips. New methods include:Export2Excel, Export2XML, SetColumnHeaderToolTip, GetCheckBoxes, SetCheckBoxes, GetHeaderWordWrap, SetHeaderWordWrap, GetHoverSelection, SetHoverSelection, GetHotTracking, SetHotTracking, GetUseExplorerTheme, SetUseExplorerTheme, GetUseTranslucentSelection, SetUseTranslucentSelection, GetUseTranslucentHotItem, SetUseTranslucentHotItem, GetShowHeaderInAllViews, SetShowHeaderInAllViews, SetFlagRenderer, GetShowItemToolTips, SetShowItemToolTips, SetColumnHeaderFont, SetColumnHeaderForeColor, SetPrimarySortColumn, SetSecondarySortColumn, SetHotItemOverlay, GetCheckStateIndices,

SetGlobalCheckState. EasyListView is a Clarion wrapper around a .NET ListView. Requirements: Clarion 6.3 / Clarion 7; ABC or Legacy Template Chains; 32-bits only; .NET Framework 2.0 and above; Windows XP and above.

Posted Monday, November 23, 2009

ClarionLive! Double Header Week

Friday: SQL Backups - different approaches by Spencer Dixon and JP Gutsatz. Saturday: Becoming the Software Vendor of Choice in your market by Rick Smith and Mark Riffey.

Posted Wednesday, November 11, 2009

Search Engine Profile Exchange November 03 2009 Release

This is a abbreviated online and web updated release - data version.

Posted Wednesday, November 04, 2009

EasyListView Version 1.02 Coming Soon, New Demo.

Changes in EasyListView 1.02 include: File explorer example; export to Excel (TPS example); export to xml (TPS example); Header Word Wrap (TPS example); Hover selection (File explorer example); Hot tracking (File explorer example); Translucent selection (File explorer example); Translucent hot item (File explorer example); Explorer theme (File explorer example); Flag renderer (File explorer example, Attributes column); Item tooltips (File explorer example); Column header font/color (TPS example); Hot item overlay (TPS example); Check boxes (File explorer example). New demo app available. EasyListView is a Clarion wrapper around a .NET ListView.

Posted Wednesday, November 04, 2009

Noyantis Codejock Wrapper Templates Now Codejock v13.2.1 Compatible

New versions of all of the Noyantis Codejock Wrapper Templates are available: 2.01 - CalendarPro Wrapper; v2.02 - CommandBars Wrapper; v1.06 - DockingPane Wrapper; v1.11 - PropertyGrid Wrapper; v1.12 - ReportControl Wrapper; v1.22 - ShortcutBar Wrapper; v2.01 - SkinFramework Wrapper; v2.01 - TaskPanel Wrapper. Modifications include: Codejock v13.2.1 compatibility added. Also, the CommandBars Wrapper tpl has a Bug Fix for a problem that was introduced in v2.01 - your app could potentially GPF in Windows XP. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

CapeSoft Clarion 7.1 Builds

For those who are brave enough to put them through the test, you can grab the binary installs of CapeSoft accessories for the C71 IDE, Except OfficeInside 3. The non-DLL products have the same install for c70 as c71 so you don't need to re-download those.

Posted Wednesday, November 04, 2009

Xtreme ToolkitPro, Xtreme SuitePro 13.2.1

This is a maintenance and support release that addresses some outstanding issues that were not addressed with 13.2.0, including feature enhancements and fixes. The 40% discount is good towards all ActiveX components and renewal subscriptions through the end of 2009. Codejock SuitePro 2009 vol.3, Codejock CalendarPro 2009,

Codejock Command Bars 2009, Codejock Controls 2009, Codejock DockingPane 2009, Codejock PropertyGrid 2009, Codejock ReportControl 2009, Codejock ShortcutBar 2009, Codejock SkinFramework 2009, Codejock TaskPanel 2009. Full license (with 30 Days Support) - XX Developer License with 30 Days Support, ActiveX COM Full license (with 1 Year Support) - XX Developer License with 1 Year Support, ActiveX COM Subscription Renewal (with 1 Year Support) - XX Developer License with 1Year Support, ActiveX COM. To use it select a product, add to cart. After that use Coupon Code (enter it into field in Cart form) CLARION and push Update/Recalculate button. You will see a new discounted price (and small (E) icon)!. After you can Checkout. Pease note that Coupon Code above not suitable for Noyantis wrapper Templates.

Posted Wednesday, November 04, 2009

Aussie DevCon Training DVDs

The upcoming video series from the Aussie DevCon covers the whole seven days of training and conference. There were four audio sources, two CCTV type video sources, one data projector output capture, plus screen capture using Camtasia Studio, plus the original PowerPoint files. These are all being mixed to give the complete professional package. The packages are being broken up into two days of C7 training, two days of Clarion.NET training, and three days of Convention presentations.

Posted Wednesday, November 04, 2009

DockingPane Wrapper Template 1.05

Version 1.05 of the DockingPane Wrapper template is available. Modifications include: Codejock v13.2.0 compatibility added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

SkinFramework Wrapper Template 2.00

Version 2.00 of the SkinFramework Wrapper template is available. Modifications include: Codejock v13.2.0 compatibility added. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

ReportControl Wrapper Template 1.11

Version 1.11 of the ReportControl Wrapper template is available. Modifications include: Codejock v13.2.0 compatibility added; Icons now automatically added to Project List (if filename prefixed with a "~"); Horizontal + Vertical Grid Styles added; Browse reload option added to Print / PrintPreview control templates; New method added - SetGridStyle; New method added - SetRowSelected. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

TaskPanel Wrapper Template 2.00

Version 2.00 of the TaskPanel Wrapper template is available. Modifications include: Codejock v13.2.0 compatibility added; Save and Restore Layout feature added; Icons now automatically added to Project List (if filename prefixed with a "~"); Font attributes added to Group definition; Font attributes added to Item definition; Markup option added to

Item definition. New Options: Groups Expandable (All Groups); Group Expandable (Individual Group); Hot Track Style; Item Layout (All Groups); Item Layout (Individual Group); Multi Column. Various new class methods have been added including: GetFocusedItem; GetGroupProperty; SetGroupExpandable; GetHotItem; GetItemProperty; SetDefaultItemLayout; SetDefaultMinClientHeight; SetExpandable; SetGroupCaption; SetGroupProperty; SetGroupItemLayout; SetGroupMinClientHeight; SetGroupVisible; SetHotTrackStyle; SetItemAllowDrag; SetItemAllowDrop; SetItemProperty; SetItemCaption; SetItemEnabled; SetItemToolTip; SetItemVisible; SetMultiColumn. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

PropertyGrid Wrapper Template 1.10

Version 1.10 of the PropertyGrid Wrapper template is available. Modifications include: Codejock v13.2.0 compatibility added; Default Display Text Lines option added to MultiLine Item; New method added - DeleteAll; New method added - DeleteCategory; New method added - DeleteItem; New method added - SetItemDisplayLines. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

CalendarPro Wrapper Template 2.00

Version 2.00 of the CalendarPro Wrapper template is available. Modifications include: Codejock v13.1.0 and v13.2.0 compatibility added; Manual creation of Schedules simplified; Additional Week / Month view options added; Customizable tooltip formats added; Markup Text facility added to events; Read Only Mode added; DatePicker Class added; Event Handler facility added to both the Calendar and the DatePicker controls; Cell Background Colour facility added; Drag and Drop facility added; Allow Events to be Moved option added; Allow Events to be Resized option added; Display Next / Previous Event Buttons option added; Display Expand Buttons CaptionBar option added; Display Multi Columns CaptionBar option added; Display on a Single Line CaptionBar option added; ImageRsc storage variable increased; Refresh parameter added to SetEventRecurring; Bug fix - CaptionBar would disappear in Timeline Mode; Bug fix - Events could be temporally duplicated if Ctrl key held down while dragging event; Bug fix - Threading Issue in c55 when Language resource used; Bug fix - DatePicker NewDate procedure was not being called when MonthChanged event detected. New procedures added: ClearCustomCellColours_????; SetCustomCellColours_?????. Various new class methods have been added including: AddDropTarget; AllowEventMove; AllowEventResize; ApplyOffice2007ThemeFile; MouseClick; GetEventBody; GetEventSubject; GetReadOnlyMode; GetEventRecurringState; GetScheduleCount; ProcessDropTargets; SetReadOnlyMode; SetEventIconAdd; SetEventIconDelete; SetEventLocation; SetEventMarkup; SetEventRecID; SetTooltipFormat; SetMonthCompressWeekends; SetMonthEventEndTimes; SetMonthEventClocks; SetWeekEventEndTimes; SetWeekEventClocks; SetNonWorkingHourColour; SetWorkingHourColour; ShowCaptionBar; ShowCaptionDateLabel; ShowCaptionExpand; ShowCaptionMultiColumn; ShowCaptionScrollDate; ShowCaptionSingleLine; ShowCaptionSwitchView; ShowCaptionTimeLine; ShowEventMarkup; ShowEventNextPrev; Reset. Redundant methods: DefineSchedule; EmptySchedules; SetRemindersObj. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email. PLEASE NOTE, For those users waiting for the ability to remove / modify an individual occurrence of a Recurring Event, then this modification will be available very soon (the code is complete but just needs adding to the TPL).

Posted Wednesday, November 04, 2009

CommandBars Wrapper Template 2.01

Version 2.01 of the CommandBars Wrapper template is available. Modifications include: Codejock v13.2.0 compatibility added; Window Reset added to NewSelection_?? procedure; New method - SetComboSelectionByText; New method - SetCtrlVisible; Bug fix - Initial selection corrected for DropLists; Bug fix - Initial height incorrectly calculated in c6.1; Bug fix - DeleteControl not deleting Ribbon Groups and Ribbon Tabs correctly. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

ShortcutBar Wrapper Template 1.21

Version 1.21 of the ShortcutBar Wrapper template is now available. Changes include: Codejock v13.2.0 compatibility added; Internal API calls renamed. The new version can be downloaded from the Members area using the original download and registration details contained in your sales email.

Posted Wednesday, November 04, 2009

RPM Update and Production Schedule

RPM for C7.0.5768 will be available for download before end of business (ET) Tuesday, 27-Oct-09. This release includes (for ABC only): Mouse Wheel support in single page and thumbnail view and the fix for a Drill Down bug when a link calls another RPM preview. All changes are documented in source as well as in the Change Log provided in the install.

An RPM install for C6.3.9051, source synchronized with the C7 release, will be available by end of business (ET) Thursday, 29-Oct-09. After the AFE install for C7 is released in early November work will proceed on a new AFE server with updated FaxMan libraries. After release, work will resume on RPM/ABC support for Fomin Report Builder and the back port to legacy.

Posted Monday, October 26, 2009

SetupBuilder 7.0 Build 2754

SetupBuilder 7.0 Build 2754 is now available. This release is available, free of charge, to all SetupBuilder customers who have an active SetupBuilder maintenance subscription plan. The update contains some important bug fixes and improvements.

Posted Monday, October 26, 2009

SetupCast 1.6.0

Version 1.6.0 of SetupCast is available for download. This is a free upgrade to all registered users. Just use the Check for Update feature or download a fresh copy. This release provides Windows 7 compatible installers and full support for automatic publishing of SetupBuilder 7 projects.

Posted Monday, October 26, 2009

Clarion Magazine

Aussie DevCon Training, Part 1

by Geoff Robinson

Published 2009-11-11

[Aussie DevCon 2009](#) was held in Eden, New South Wales, Australia this year from 12th to 18th October. Eden had proven to be a popular location for the conference last year and once again three SoftVelocity staff made the trip out from USA to attend.

About 40 people attended Aussie DevCon this year with about 25 doing the training which preceded the convention proper. The training was titled "Exceptional Development Techniques in Clarion 7 and Clarion.NET".

Bob Foreman took the training for the first couple of days. He started by introducing himself and Pierre Trembley. Bob has been with TopSpeed then SoftVelocity since 1989 when he initially started in technical support. In the early 90's he moved to education/training and in the late 90's consulting. Since around 2000 he has been involved with documentation. Pierre comes from near Montreal in Canada and has been with TopSpeed/SoftVelocity since joining the consulting group in 1999.

Clarion 7.1

Bob stated from the start that the *goal* was to have C7.1 in our hands by the end of the conference (we were in fact each given a pre-release copy of C7.1 on the last day of the conference.)



Figure 1. Bob Foreman conducting training

We did the usual round of brief introductions and it is always great to see a mixture of old familiar faces from

numerous previous conferences plus new people we haven't previously met. A show of hands indicated about half had been at the previous conference in May last year.

Bob suggested that many seasoned Clarion users ignored the *Getting Started* and *Learning Clarion* manuals — after all they already know all that stuff. Bob pleaded with us to read them for C7 as they have been rewritten with lots of new information.

He stated that all the code in C7 had been refactored for speed and that generating, building and compiling are faster in C7 than C6. The use of the MSBuild system was an advantage.

Bob was using a version 7.1.6085 (debug). This compared to 7.0.5768 on my machine. In fact Bob had what he referred to as a "composite environment" which also included Clarion.NET 2.0.6085 (note the title bar in Figure 2. This is different to us as we currently have separate environments for the two products, but this will come to us eventually.

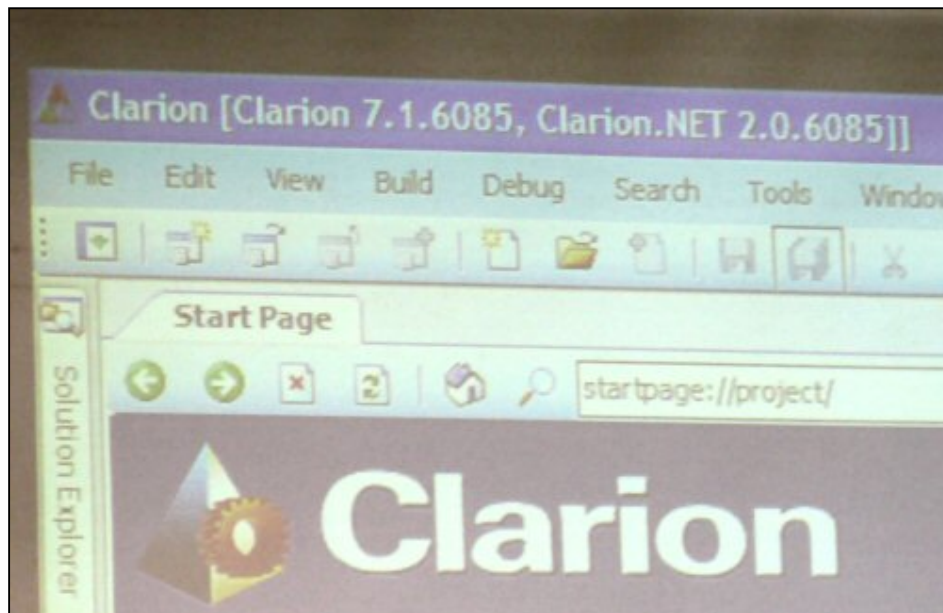


Figure 2. The combined IDE

Bob mentioned that everyone will have their own preferences with regards to layout of the various IDE pads (windows). The pads can be pinned but by default are unpinned. When pinned the pads slide out of the way; you just see the tab until you mouse over them. This frees up your workspace as there are something like 17 pads.

If you wish to go back to the default setup you can go to the Windows Explorer command line and type

```
%appdata%\Softvelocity
```

This brings up the Clarion folder which you then delete. I recommend a bit of due diligence before you do this, and you may want to keep a backup of the Clarion folder just in case.

Bob pointed out you could use the address bar on the start page of C7.1 to go to web sites etc. I thought "I normally use a browser for that kind of thing"

IDE options

Bob showed that some IDE menus are static while others are dynamic and change with context. Some are conditional — you can see them but they might be greyed out if not relevant in the current context. There are also lots of pop-up menus which appear when you right click in an IDE window. The pads are dockable in the IDE and you can always use Control-F6

to find windows if they seem to go AWOL.

We went through lots of options for the IDE which are under Tools -> Options, including the UI language. Linus, who was out at the conference from Lithuania, did not have his language there — Bob pointed out that he could add it himself using a configuration similar to XML but Linus was comfortable with English so this was no drama.

There is an option to run builds as separate processes rather than threads which can apparently make things faster.

You can alter the level of detail for output — minimal, normal or detailed. Detailed is handy if something goes wrong as you can pinpoint where it was up to.

The editor does not yet have macros but it does have code snippets which can be useful. One user at the conference complained that when he loaded a new version of C7 it blew away all the code snippets he had set up — so be sure to run a backup first.

Bob mentioned that he likes to have Automatic Indentation on in the text editor — this is a personal preference type of thing. (As an aside I remember at a much earlier conference two people sitting at the dinner table arguing over whether you should indent two characters or three. As with all religious wars there is really no "correct" answer but each side fervently believes theirs is the one and only true and righteous path. And many feel the need to convert the world to their view. They turned to me for some sort of adjudication and I upset them both by replying "Well actually I use four characters....")

Bob showed us how we could easily add external programs to the IDE by adding the new Report Writer. This product is currently in Beta and will come with 7.1. There was a session on this later in the week so I will hold off more detail until then...

One window designer feature Bob pointed out was "Smart Tags". These look like a small VCR "forward" button and appear when you highlight a control. Click on the smart tag to bring up further options.

We next looked at the property toolbox which Bob mentioned he wished was on by default. To display the Properties Pad, you select View -> Properties from the IDE Menu, or you can just press F4. I guess the nearest equivalent in C6 was pressing F12 on a window and being able to see and adjust properties for each control.

You can run different versions of Clarion within the IDE. Bob mainly uses C7.1 these days but occasionally fires up the C6 compiler in the new IDE (still using the C7 AppGen) where someone complains that something used to work in C6 but doesn't in C7. When asked whether we could run multiple versions of C6 in the IDE Bob responded "Yes, absolutely". There was talk of Macs — I got excited - Clarion for the Apple Macintosh? — alas no, we were talking of MACS "Multiple Available Compiler System".

Applications General Options apply to all applications. This is in contrast to Application Options which relate to the currently opened app.

There was an option to cache an app in memory after editing. This speeds up loading etc but is not recommended for large multi-DLL systems unless you have got lots of memory to use up.

Migration

The next topic we covered was migration. Migration could be divided into three stages — Conversion, Generation, and Build.

Before you start conversion of an app you need to make sure you have all relevant templates registered. I think Bob mentioned that in an earlier version if a template was not found it was just removed but this has been fixed now.

Also don't forget to install all your add-ons like the IP driver, IMDD etc. (I have written down that there was a free upgrade to these from C6 to C7 but don't flame me if I misheard that!)

Start with a small simple app and work up to your more complex systems — do not start with an enormous multi-DLL

system first off.

The dictionary converter will clean a lot of things up — many things that should not have been allowed previously but which you managed to get away with are now not allowed.

Bob was very encouraging: "Do not be afraid of C7 — convert over now!"

When an application or dictionary is to be converted you are presented with a message warning you that conversion is about to happen. Once converted to C7 format they cannot be used back in C6 or earlier. This applies even if you are using the C6 compiler in the new IDE. Once converted the application is displayed in the Solution Explorer along with associated source files, file drivers, icons and images. There is also an Applications Pad which has a toolbar at the top to do generation, building and running the application.

The C7 project system uses MSBuild which comes with the .NET framework. View the C7's online help to see various build options. Solution files (*.SLN) are used to store C7 projects. (Clarion Magazine also has a [number of articles](#) on MSBuild.)

We had a look at a multiple DLL solution. You start by opening the app for the .EXE then after that is converted you get the option of adding other app files that are called from the EXE. You can either add the other apps manually (via Add - > Referenced Projects) or let the system do it automatically for you.

There is a dependency editor which allows you to "view all applications in the active solution and view dependencies and the order that the applications are built". There was quite some discussion about [circular references](#) between DLL's and the fact that C7 does not readily permit them. The upshot (as far as I was concerned anyway) was that if you have circular references (where DLL A calls B and a procedure in B also calls one in A, or perhaps A calls B which calls C which has a call back to DLL A) then you should remove them. A fair bit of work (I went through the exercise myself some time back) but worth it.

With regards to the Project System Bob posed the rhetorical question "What has changed?" and answered that it was better to ask what had *not* changed.

There was some question re the difference between the Build option and the Clean option. The answer was that the Clean option deleted OBJ files. In fact it does a bit more than that — by default a solution has three sub folders (BIN, MAP and OBJ — folder names can be changed in the redirection file). These then have Debug and/or Release depending on your settings. Doing a Clean removes all files from these subfolders.

Sometimes when doing a conversion, a window or report may have errors that need to be fixed. These are highlighted with an icon with an exclamation mark "!".

Various settings for projects can be found in the Project Properties Application tab. Output type (exe, LIB or DLL) was straightforward whereas Link Mode (DLL, LIB or OwnDLL) is a little different — this was C6's *Memory Model*.

My understanding is that DLL will create an exe that calls all other DLL's (similar to "Standalone" in C6), LIB will link in the other functions and create a stand alone exe (similar to "Local" in C6 — and yes it *is* confusing), and finally OwnDLL will use your DLL's but not the standard Clarion DLL's). Our training guide mentions that "This means that all Clarion functions must be included and exported from another user created DLL". To be honest I'm not sure why I would use this OwnDLL option as it sounded too much like hard work but perhaps I was misunderstanding it.

In C6 we have the ability to run a command after a compile ("Programs to Execute" in the Project Editor). C7 also adds a new "Pre-Build" event command line. I gather from Bob that this came free as part of the MSBuild system.

A handy new feature was the ability to copy referenced DLLs to the output directory — these are the files that you need to deploy with your app. These are automatically refreshed when you do a build, so you will always have the latest versions.

We had a look at the Dictionary Explorer and Bob said that 90% of things can be done from Quick View — the exception being triggers where you need to use the Entity Browser.

The Output Pad gives you information when you do a build (eg. Starting, completed etc) and is a simple text window, while the Error Pad is more sophisticated, categorising output into Messages, Warnings and Errors. As with earlier versions, double clicking on an error takes you to the relevant source line. You can choose between three levels of output — minimal, normal and detailed. There are also two further levels reserved for future use.

There were some general DLL hints:

- It is better to have a few big DLLs rather than numerous small ones as they will load more quickly.
- Make sure your DLLs are in the same directory as the exe so the operating system does not have to go hunting for them.
- Place related code and data in the same DLL. (I was not sure about this one — I imagine most Clarion developers have a data DLL).
- Avoid cyclic dependencies (already mentioned above). There was some discussion about using the `init_priority => n pragma` which I think was to set the order of calls to try to stop problems caused by cyclic dependencies. As I said — it's simpler just to avoid cyclic dependencies.
- Declarations must be the same in all apps (with the addition of External and DLL attributes).
- In the case of Files, only one declaration should not have EXTERNAL and DLL attributes — and this is the only one that should have the OWNER and NAME attributes. Ditto for Queues, Groups and Classes.
- For large apps a good number of procedures per module is 10 to 15. Compared to having each procedure in its own module, this will make source generation take longer, but this will be more than made up for with quicker compilation and linking.

The dictionary editor

There was some time spent on the dictionary. As most Clarion developers know, the more time you spend up-front setting up your dictionary the better. Make sure you have an auto-incrementing LONG as your primary key on each table.

Column Pools are handy — you set up a standard declaration for commonly used fields to ensure consistency between tables. You then use the "Derived From" column option in the dictionary.

The dictionary now tends to have incremental locators rather than just step locators which only use the first character.

In C6 you could sort your tables in either Alphabetical, Date Last Modified or Original Order. In C7 "Original Order" is renamed "Creation Date" but I guess it is the same thing.

Client side triggers are set up in the dictionary and can be executed before or after a table entry is inserted / modified / deleted (ie. there are six different trigger types). They were first introduced in C6 and are currently only supported in ABC. Triggers can be used to enforce business rules, keep audit trails (log events), derive column values or store complex conditional or computed fields etc.

We looked at the Synchronizer and Table Import Wizard. The latter is similar to the Synchronizer but used when you want to import a table for the first time. The Synchronizer doesn't back up data so make sure you have a copy of your data and your dictionary before starting. Bob pointed out that while many tend to think of the Synchronizer only as an SQL tool, it can in fact be used to synchronize any dictionary with any file format. It is especially handy for multi-table imports.

We had a look at the data browser which is similar to the old TopScan (but not limited to TPS files) and the data diagrammer. The diagrammer has the ability to add or remove tables from the view — essential I would think where you have hundreds of tables in a dictionary and only want to view a subset.

The diagrammer can also show table properties and statistics. It has various zoom features and the ability to print to pdf or image formats. On the Diagrammer Toolbar there is a button that opens up a separate tab for a Report View which gives "a powerful visual comparison of key elements in your dictionary" and helps you find any discrepancies in your dictionary. It does this by listing all the fields alphabetically regardless of which table they are in. So, for example, you might see that

you have an address field 30 characters long in one table and only 20 in another. Of course if you were using the Column Pools you would not have these problems! You can click on the column headers to change the sort order etc.

Some users felt the diagrammer was not yet as powerful as what we have currently as it does not allow you to alter things diagrammatically and then automatically update the dictionary. I must admit I have never done that but it goes to show that we each tend to work in our own particular (peculiar?) way. I mean some at the conference were surprised when they saw me using the Ztree file and directory manager — doesn't everyone use that?

Recovery

A great new feature of C7.1 is the new recovery ability. A transaction file (.~app) of all changes since the last save are kept and in the *unlikely* event of an IDE crash you can recover to exactly where you were. In the Application Options you have the ability to specify how large the recovery file is allowed to grow before doing an auto-save.

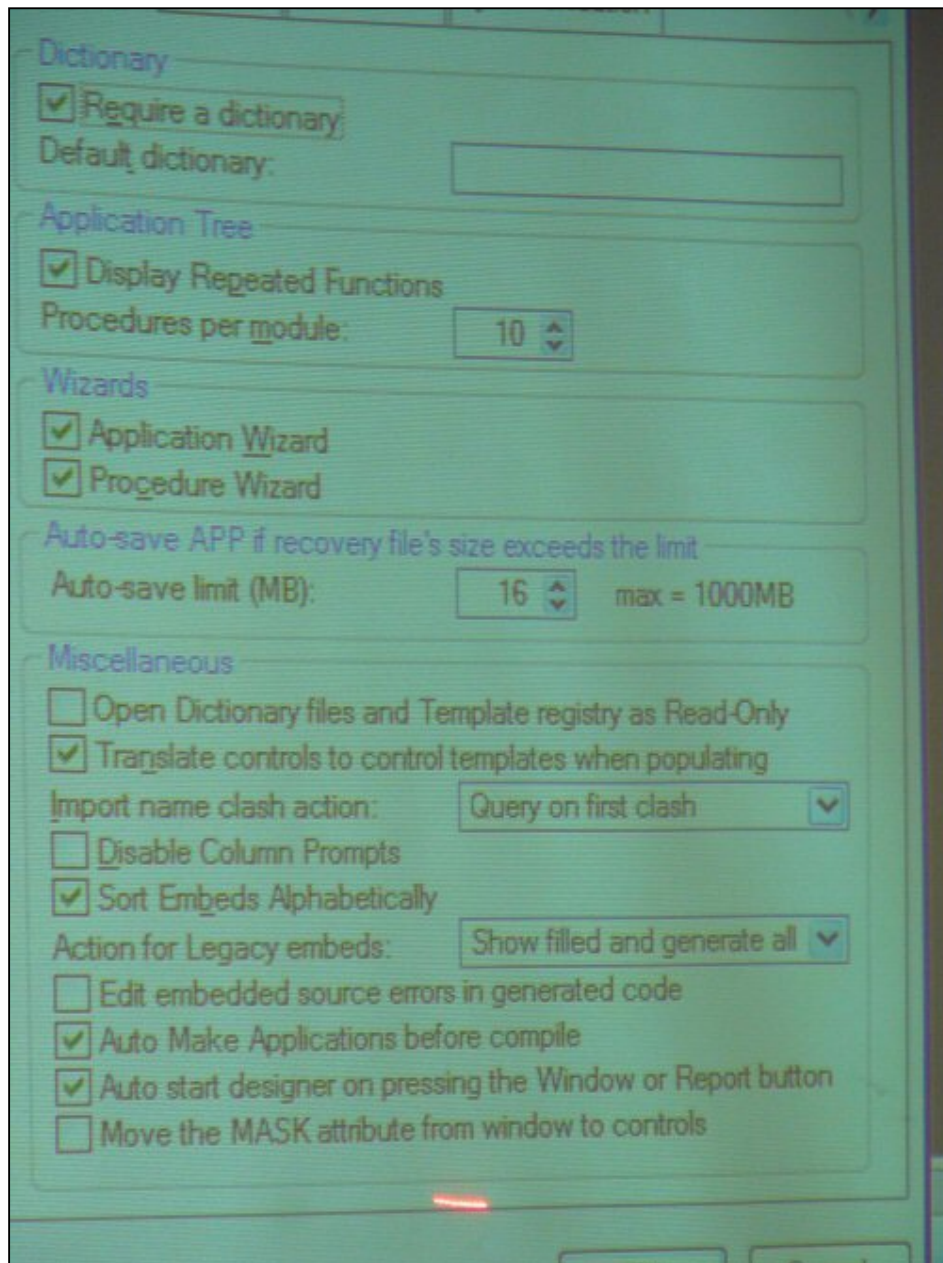


Figure 3. New application options

I must admit to a slight chuckle at the notes on this new Application Recovery ability which never entertained the possibility

of an IDE crash but spoke instead of protection from malevolent "outside forces" that could cause a shutdown like "power outages, disk drive issues, and even a virus attack".

In fairness Bob did demonstrate this feature by going to Task Manager and "end tasking" Clarion and it is great that with C7.1 we should not lose any work no matter what the cause of the problem - so we should not be churlish!

There is quite a lot of control over the Embeditor, including options to use it by default rather than the Embed Editor. Note that "Embed Editor" (or *Embedded Source Dialog*) relates to editing one individual embed point whereas "Embeditor" (or *Source Editor*) means editing that embed point in the context of the surrounding code.

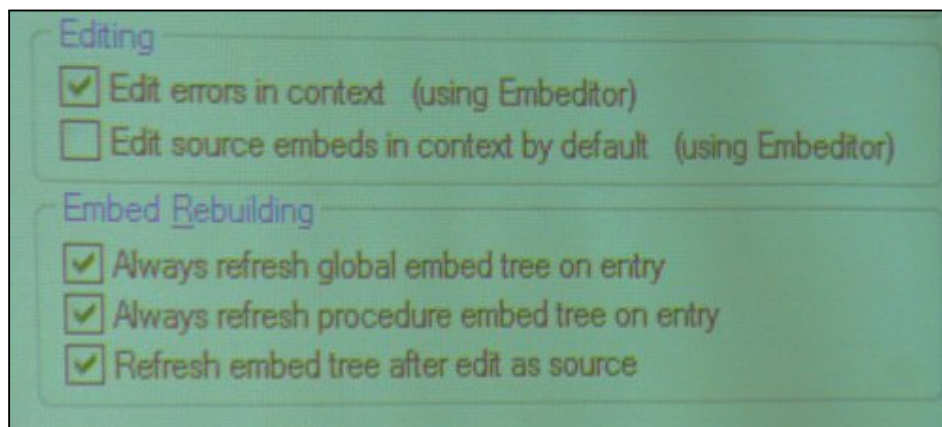


Figure 4. Embed options

We looked at the redirection (.RED) file which controls your search and storage paths for applications and projects. The RED file can be tailored for any particular application. C7 first looks for a red file in the application directory and if not found then uses the default redirection file in \bin.

We had already covered a lot of ground and it was only lunchtime!

[Read Part 2](#)

[Geoff Robinson](#) lives near the beach in Melbourne, Australia, and is an active member of the local Clarion User Group. His company, [Vitesse Information Systems](#), specializes in software for local government. Geoff was impressed by Clarion back in the DOS days and grabbed the early betas of Clarion for Windows when they first became available; he has been using Clarion as his primary development environment ever since. When not in front of a computer Geoff enjoys listening to music, singing bass in a local choir, and spending time with his three young children.

Reader Comments

[Add a comment](#)

Clarion Magazine

Aussie DevCon Training, Part 2

by Geoff Robinson

Published 2009-11-11

[Read Part 1](#)

The meals at the Fisherman's Club were generally very good and you could not get fresher sea food. I think I've had my supply of mussels for the year...

Wizards and themes

After lunch we had a look at Wizards and Theme control. We generated a C7 app using the Wizards, much in the same manner as with C6. Note that in the case of the application wizard, the browses and forms generated take their settings from those you set up for the more specific Browse and Form Wizards. Wizards are essentially special utility templates.

Many of us will remember the *wizatrons* from some time back — I remember seeing a video presentation David Bayliss did some years back at a DevCon. Anyway while the wizatrons are largely extinct their legacy lives on in the form of the Themes — which Bob said had comparable power but were "much easier to use". Themes are groups of settings that control the template settings. They are simply text files similar to an ini file and their location is set in C70TFT.INI (which is in the Clarion bin folder) but by default they are in %ClarionRoot%\Template\TFT. A theme maintenance wizard is a utility that writes to a theme file. There are two categories of themes : visual customisation and prompts for template actions.

In the past I have rebased my DLL apps based on information Carl Barnes provided in some [very helpful articles](#). This has now been made easier with inclusion into C7 of the ability to specify the base address of each DLL. The idea is that each DLL has its own base address with enough "breathing room" in terms of memory so that DLL's don't "clash" and therefore load more quickly. If a DLL cannot load at its preferred memory location it needs to be relocated by the loader to a different address and all memory pointers must be adjusted accordingly. Not only does this take time but it means the DLL cannot be shared. Support for rebasing is built-in for apps and hand coders can modify project settings (see the IMAGE_BASE help topic).

Icons in C7 can now be of various sizes (with multiple instances/sizes per ICO file) — apparently in earlier versions they had to be 32 x 32.

There has been more work done in C7 on merging the ABC and Clarion templates. The Clarion templates are no longer referred to as "Legacy" templates as they continue to be improved to support more of the ABC libraries — "many of the foundation classes of the ABC chain are now a part of the Clarion chain".

UI improvements

One welcome feature in C7.1 is that window and reports now have separate buttons to get into the designers with the familiar "elipsis" buttons to go into the equivalent source code — just as with earlier versions of Clarion — rather than tabs. And the windows now have a window preview button. SoftVelocity had originally thought this was unnecessary but were convinced otherwise by the desperate pleas of users. "This proves SoftVelocity listens!" Bob proclaimed.

The new C7 designers have three main components — a control box for populating controls, a properties pad and the designer workspace which has a toolbar for control alignment, tab order and so on.

The toolbar on the properties pad allows sorting alphabetically or by category, while the worksheet allows you to change properties much like the F12 editor in C6.

Populating, positioning and resizing controls all seemed very familiar but alignment of controls seemed to be opposite in behaviour compared to earlier versions which might take a little readjustment. In other words controls are aligned with the FIRST control that was selected and not the most recently selected control. The first control selected (the *anchor* control) has white handles whereas the later controls have black handles so it should be clear enough...

The new previewers (window and report) now show dummy data which can be controlled by you via XML format files which have the extension *.winprev. You can specify different data to display for different column names — so you would have different data for Given name as opposed to Surname.

The final part of Bob's presentation on the first day of training was entitled Browse Expertise and was a collection of new (ish) features combined with tips and tricks.

We looked at the "Do not include Primary Key in view" option which is used when working with SQL tables. By default primary key components of the table are automatically PROJECTed into the view and this option stops that in order to allow valid GROUP BY SQL statements to be generated.

Having just mentioned that Clarion templates now have much of the ABC functionality, sorting of headers is an exception — this option is currently ABC only. I guess this also brings up a point about third party versus standard Clarion templates as many third party templates had column sorting for years before it was introduced into the standard templates.

This was in fact a question later in the week at one of the panel discussions, and Bruce Johnson from Capesoft reiterated his position that he thought SoftVelocity should in no way be put off adding new features for fear of upsetting third party vendors. The core product needs to constantly improve and if that now covers an area previously provided by third parties then that will push the third party providers to further improve their own offerings.

Browse expertise

There is an option to disable pop up support on browses, although I can't see myself ever using that. Just as I said earlier that different developers like doing things differently, so it is with users. The more options they have the better. Having said that I guess if developing for a naïve user you might want to limit their choices to save confusion.

There is now a global option for Auto-Sizing Browse box columns where the user double clicks on the right line of any column to get its width to adjust to the widest entry currently being displayed. This can be disabled at the procedure level for the current browse.

There was some discussion on the List Format Manager which enables users to re-adjust the column formats of a browse list. The main complaint was that it was not very practical as each user does not get to save their own particular format but shares the one format which by default is stored in a topspeed file. So one person could twiddle it (technical term) to their

own requirements only to have someone else come along and change it completely.

Having said that, I just looked at the format of the List Format Control File (LFM_CFile) and it does in fact have a user identifier so perhaps the above complaint was the result of a misunderstanding and hence unjustified.

```
LFM_CFile FILE,PRE(CFG),CREATE,DRIVER('TopSpeed')|
    ,THREAD,NAME('Formats.FDB')
key_Main KEY(+CFG:ProcId,+CFG:UserId,+CFG:CtrlId,|
    +CFG:FormatId),OPT,NOCASE
Record RECORD,PRE()
ProcId SHORT ! Procedure identifier
UserId SHORT ! User identifier
CtrlId SHORT ! Control identifier
FormatId SHORT ! Format identifier
FormatName STRING(30) ! Format name
Flag BYTE ! Default/current flag
Format STRING(5120) ! Format buffer
VarLine STRING(2048) ! Variable buffer
END
END
```

BrowseNoRecords is a new button control template that is only active when there are records in the browse. You can choose to have it disabled or hidden when the browse is empty.

Another button control template is the SortOrderButton. When populated it hides the tabs and instead provides a drop down list of sort orders. This is very similar to SortOrderDrop which just provides a drop list of sort orders without having to press a button.

We had a look at lots of other things: Additional Sort Orders for Browsers and the new Assisted sort type which prevents you from accidentally entering an incorrect sort string which would cause compile errors. The Higher Key Component saves you manually having to set the higher "unchanging" part of your key in your code. Clarion 7 provides better variable filter support in the Clarion template chain. Then there was Conditional totalling — you can add up your totals differently depending on conditions you provide.

I must have been getting tired as my notes by this time of the day were becoming somewhat brief and, dare I say it, cryptic. There was more you could do in regards to browse styles than I had realised. You can set up a style consisting of colors and fonts and then apply that style to a given list. The styles can be assigned either at design or run time and the templates support either method. All columns in a browse can share a common theme or different columns can use different colors. And you can make the colors conditional depending on the data, e.g. red for overdue accounts.

Then there was edit-in-place (with a new "greatly improved interface") and Advanced QBE. The Query Center has an easy drag and drop interface and allows complex expressions to be built and saved. Query Center Translation allowed for translation into foreign languages as well as opening the door to natural language queries.

Forms expertise

We had covered a lot and that was only Browse Expertise.... next was Forms Expertise. Again many of the features had in fact been added back in C6 and then brought over to C7. The idea of many was again to eliminate having to embed code that was necessary previously. Things like visually showing which control you are on and field navigation (replace hitting tab with the old DOS-style ENTER key).

A show of hands indicated no-one used the Alternate Forms (VCR) style which I think is more typical of MS Access programs where you do away with a browse list and just have a form with a VCR control for navigation.

The new BLOBInControl extension template is another example of a template being added to save you writing some embedded code.

Manifests

We next discussed manifests and support for Vista and Windows 7. Vista manifest support was added in C6.3 "to interface with C7" and targeted an application's security (whereas XP manifest is primarily visual). The manifest identifies an application to Vista and allows administrators to define what security credentials it has. These manifests "can and should be signed" and set different execution levels:

asInvoker — removes Vista popup

highestAvailable — level up from asInvoker

requireAdministrator — registry access

UIAccess — for accessibility utilites

C7.1 adds Windows 7 support and targets the compatibility section of the manifest file.

Next up was Visual Styles which was introduced in C6.3 to interface with C7 and was initially an extension but now fully integrated into the latest versions of C6 and C7.

XML and NetTalk 5

Finally we looked at the options for working with XML in Clarion 7.

Bruce Johnson completed the day with a presentation on [NetTalk 5](#). This includes some general improvements as well as lots of Web Server improvements including the use of jQuery. This was broadcast on ClarionLive! and you can see Dave Harm's detailed write up [here](#).

[Geoff Robinson](#) lives near the beach in Melbourne, Australia, and is an active member of the local Clarion User Group. His company, [Vitesse Information Systems](#), specializes in software for local government. Geoff was impressed by Clarion back in the DOS days and grabbed the early betas of Clarion for Windows when they first became available; he has been using Clarion as his primary development environment ever since. When not in front of a computer Geoff enjoys listening to music, singing bass in a local choir, and spending time with his three young children.

Reader Comments

[Add a comment](#)

Clarion Magazine

IP Driver Notes

by Dermot Herron

Published 2009-11-23

Last year I wrote an article titled [IP-Enabling An Existing Application](#). Since that time I have learned more about the IP driver from several other developers on the Clarion newsgroups. In this article I'll present some of my notes.

Specifically, this is a checklist for making sure the IP_People and the ipreq.cfg files work reliably. In everything below, you must substitute *your* file names and references.

1. Write a batch-file such as CpPeop_.bat (keep the left-hand side of the filename to eight characters or less because batch files are DOS programs) containing the line:

```
COPY IP_people.dll c:\clarion6\ClarionDataServer
```

2. In the IP_people.app in the Project Editor, add this file to the ProgramToExecute. Then, whenever you compile IP_people.app the DLL will be copied to the ClarionDataServer subdirectory automatically. (And you won't find yourself working from an old file and wondering why the change you made doesn't show up!)

Do this for each dictionary that is ip-enabled! Also make sure *all* demo and testing IP_...DLLs are removed from the ClarionDataServer directory so *only* the DLLs you need for production are in there (I'll tell you why later).

3. In the ClarionDataServer directory run IPSRVMGR.exe (Figure 1). Do the following:
 - a. Remove Service
 - b. Delete Config File

c. Install Service

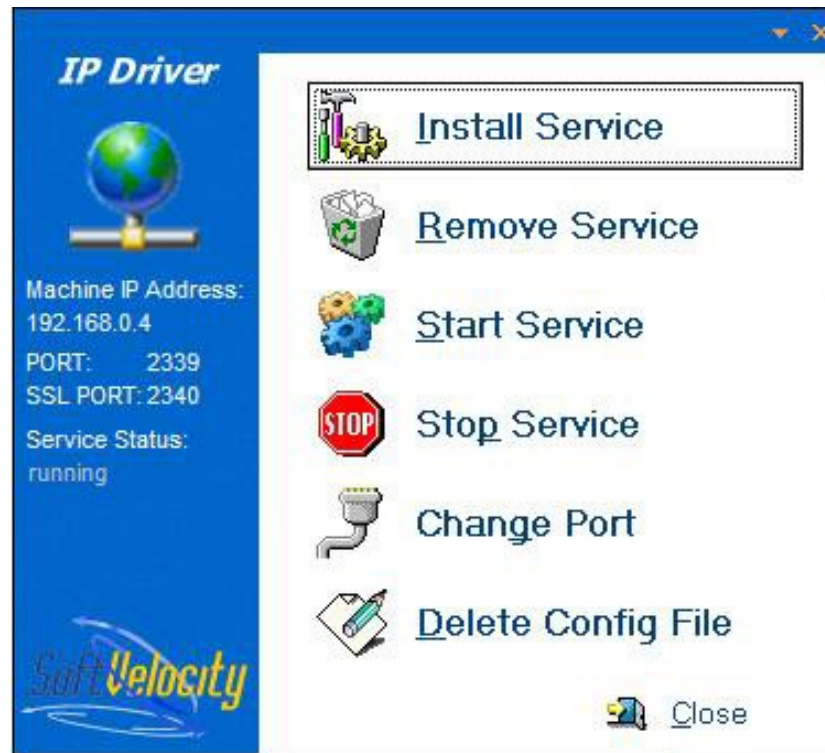


Figure 1. Running IPSRVMGR

The three steps above stop and remove the service from your working directory and restart the service in the ClarionDataServer directory with a clean, empty IPREQ.CFG file

4. Change to the RMAAdmin directory and run radmin.exe. Do the following:
 - a. Click on the chain icon and register as administrator, no password. You should get something that looks like Figure 2.

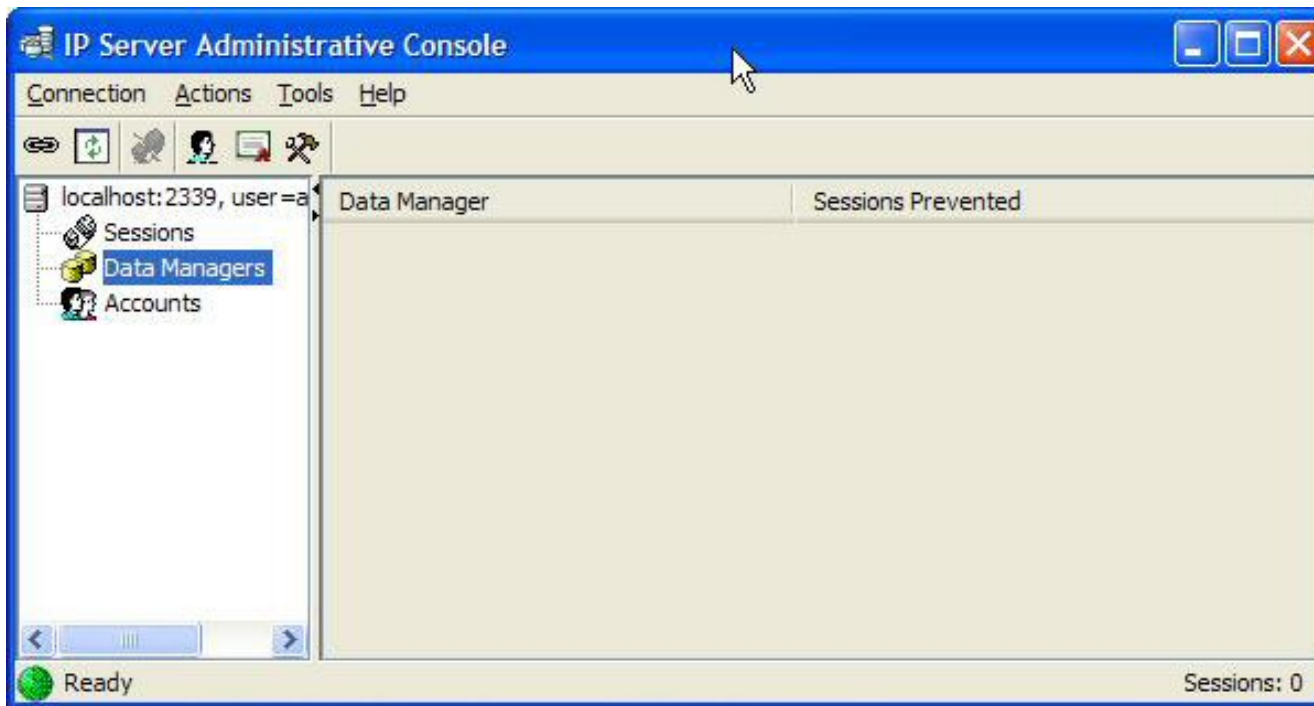


Figure 2. The administrative console

If there is anything under Data Managers then the ipreq.cfg file actually being used was not deleted properly.

- b. Right-click on Data Managers and select Register Data Managers and you should see something like Figure 3.

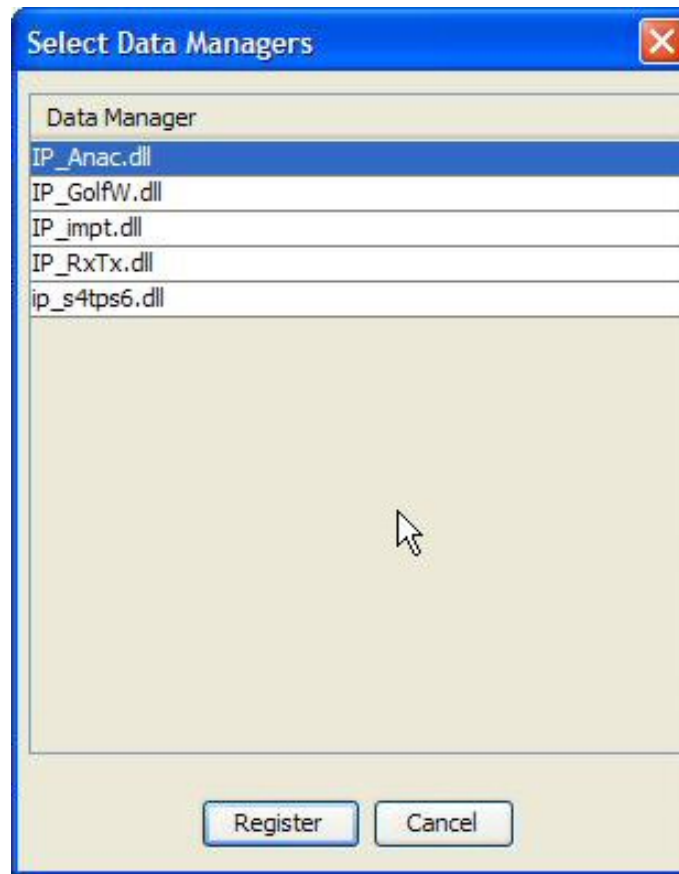


Figure 3. Registering a data manager

- c. This is where getting rid of tests and demos makes it easy. Push Ctrl-A and all will be selected. Click Register and you should see something like Figure 4.

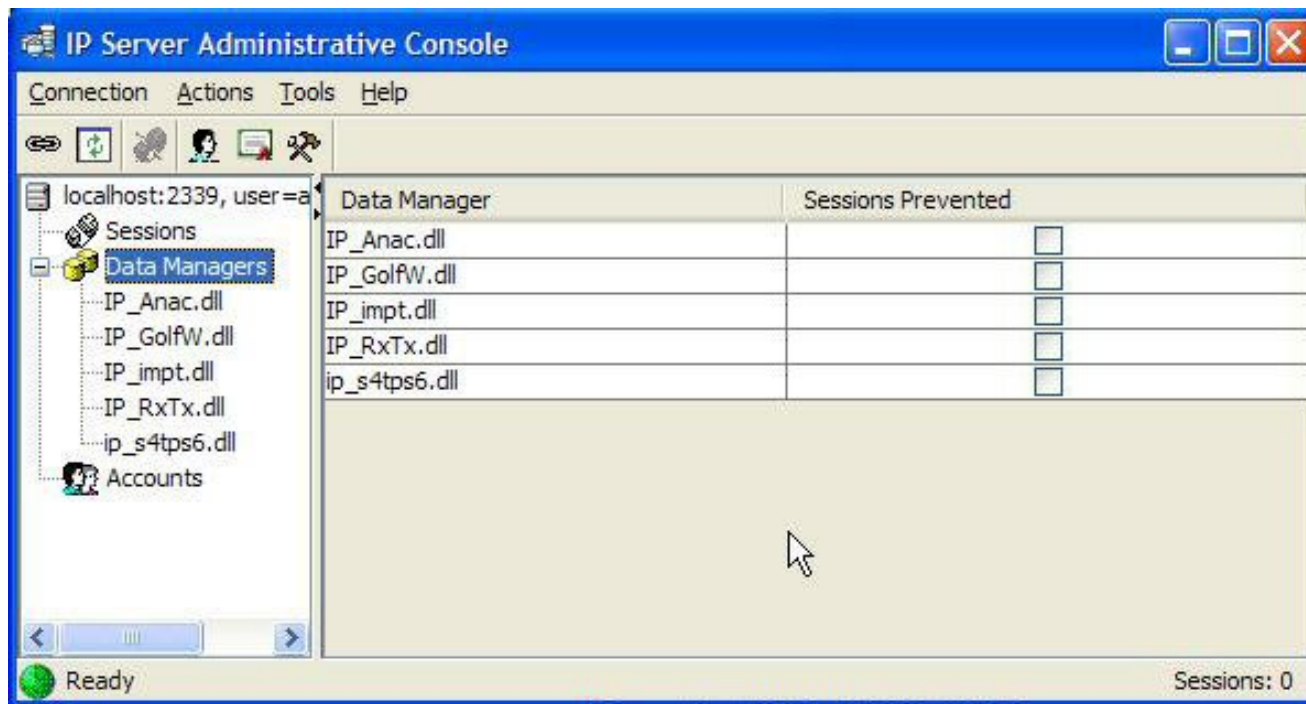


Figure 4. Viewing the registered data managers

- d. Then select Menu/Tools/Server Settings (Figure 5).

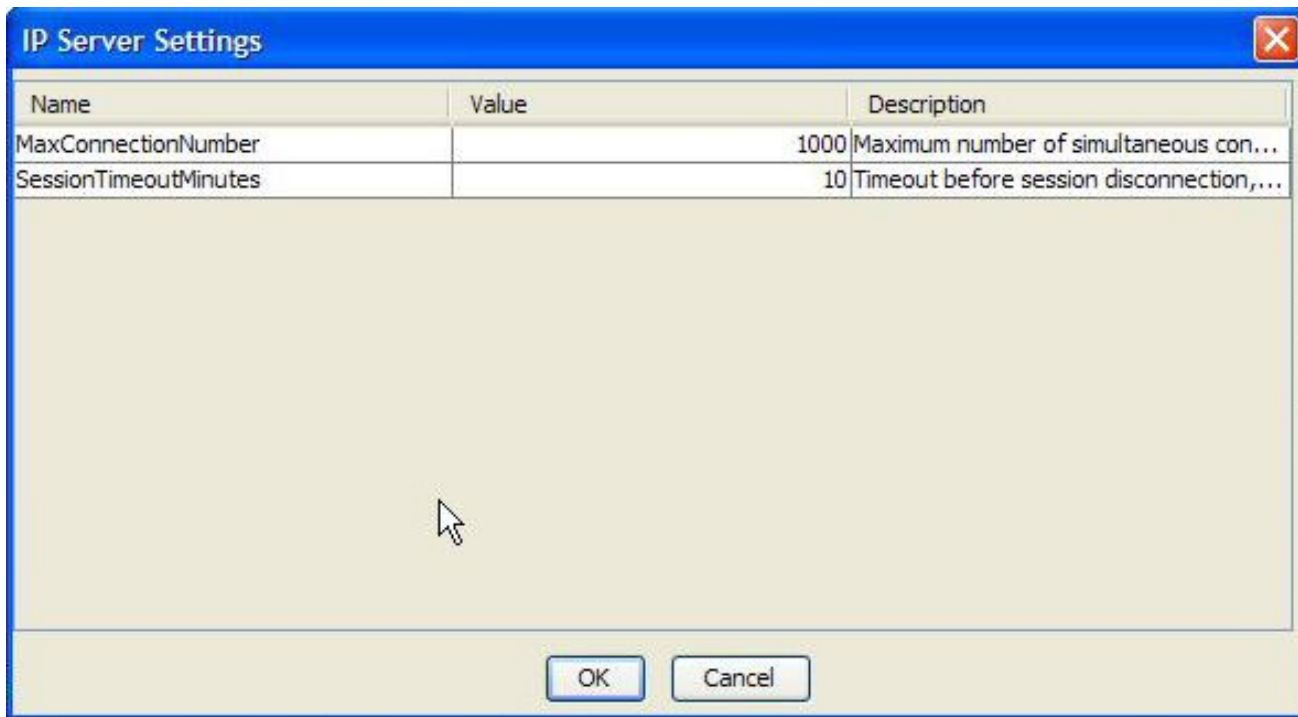


Figure 5. IP Server settings

- f. Double-click on the SessionTimeoutMinutes value (10, edit-in-place) and make it 10000 to prevent anything timing out. (bigger numbers don't store). Click OK.

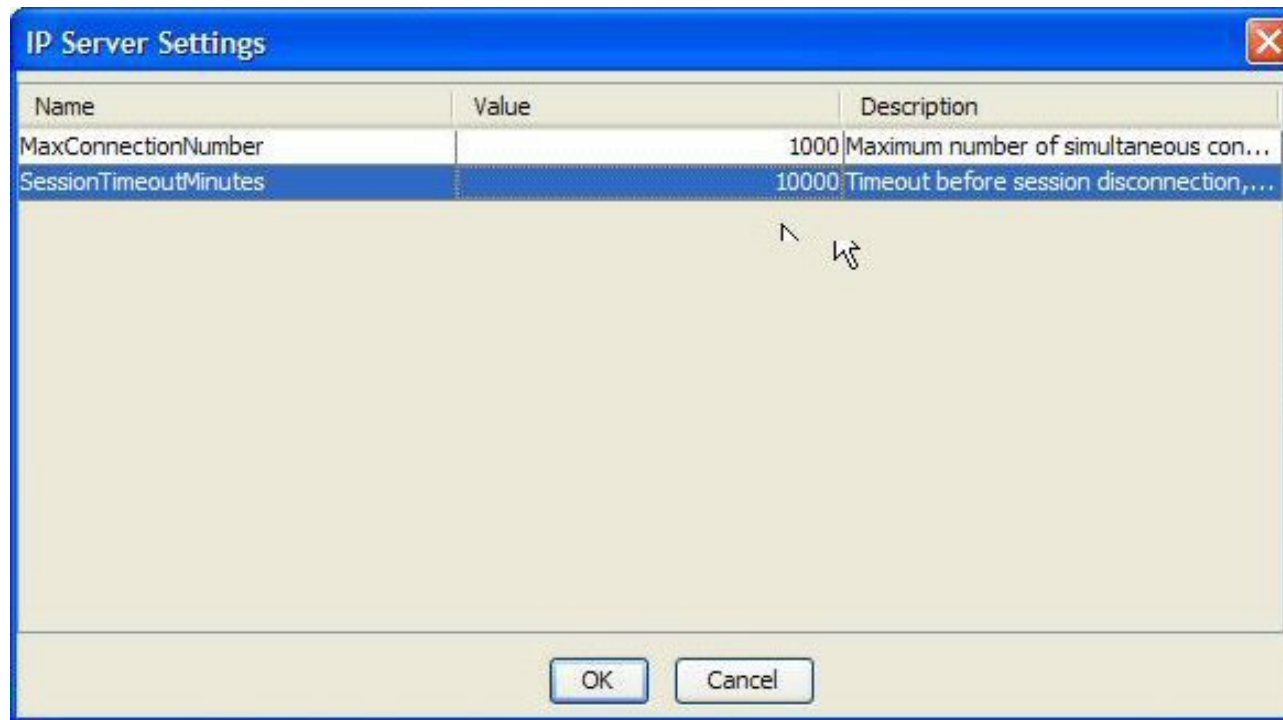


Figure 6. Changing the timeout value

Close rmadm and copy the new IPREQ.CFG to each of the subdirectories where it may be needed. (I wrote another batch-file to do this to prevent mistakes.)

Then change to your current working directory and run run IPSRVMGR there. Do RemoveService and InstallService (do *not* run DeleteConfigFile!) to make the IPDriver run where you are currently working and use that IPReq.cfg.

(InstallService uses the IPREQ.CFG in the subdir it is run from and it expects to find the data files there or in a subdir below)

Using SetupBuilder

Now for the installation of the IPDriver using the full version of Friedrich Linder's [SetupBuilder](#) (you need the scripting capability).

For the IPDriver to function properly you need all the files in the installed ClarionDataServer subdir that are not your files other than your DLLs. These include:

- claipdat.exe
- IPReq.exe
- IPSRVMGR.exe
- claipsrv.ini
- Ipsrvmgr.ini
- server.pem
- IPDSLICENSE.txt (legal)
- Ipreq.cfg
- Your DLLs

I get these directly from ClarionDataServer. Don't leave anything out — I have paid dearly with days wasted time for leaving out claipsrv.ini, as the default ports are not the ones in the ini file!

In addition you need these files from the BIN directory:

- ipds_libeay32.dll
- ipds_ssleay32.dll (even if you are not using secure-sockets)
- OSSLWrap.dll

Before you can install any new IPDriver DLLs the service must be stopped, and it must be restarted when the installation is complete. In my original article I suggested you run ipreq.exe -u to uninstall and ipreq.exe -i to install. There is a *much* better way, thanks to Chuck who answered my query in the softvelocity.clarion.databasedrivers newsgroup at news.sofvelocity.com.

The name of the service is *really* important and has to be *exactly*

SoftVelocity IP Data Server

If you don't use this name, then the service will register and run but IPSRVMGR will not be able to start and stop it, which makes trouble-shooting and hand-operation difficult.

In the SetupBuilder script at the beginning you need the three lines in Figure 7.

```
26 stop the service
27 Stop Service "SoftVelocity IP Data Server"
28 Remove Service "SoftVelocity IP Data Server"
29 Sleep for "3" seconds
30
```

Figure 7. Stopping the service

StopService and ReMoveService are EditService script options while Sleep is another script option.

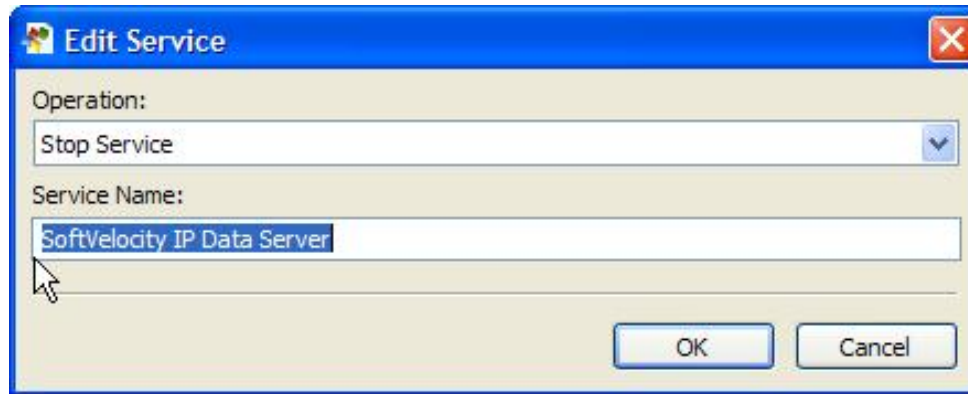


Figure 8. The Stop Service script option

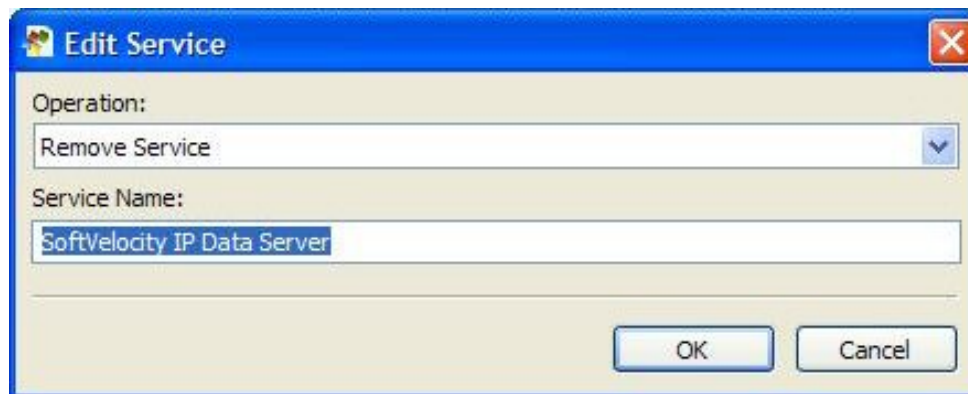


Figure 9. The Remove Service script option

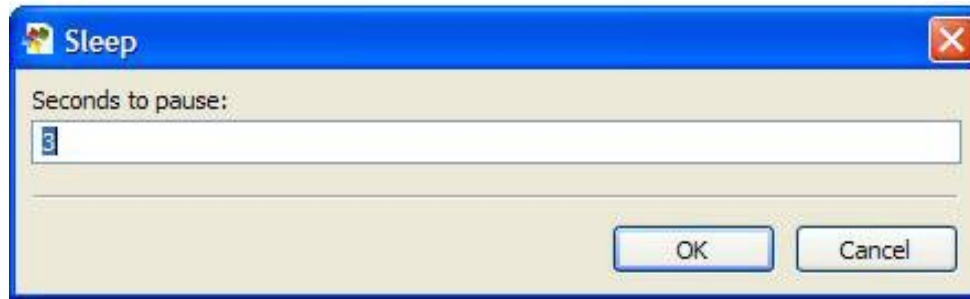


Figure 10. The Sleep script option

The Sleep option gives the operating-system time to finish stopping and removing the service.

When all the files are installed, then you have to re-install and start the service by adding the lines shown in Figure 11.

```
424 | start the IPreq service again
425 | Create Service "SoftVelocity IP Data Server" (Always Install)
426 | Start Service "SoftVelocity IP Data Server"
427
```

Figure 11. Reinstalling and starting the service

To do this you use the CreateService script option:



Figure 12. Creating the service

And again an EditService to Start it (Figure 13).

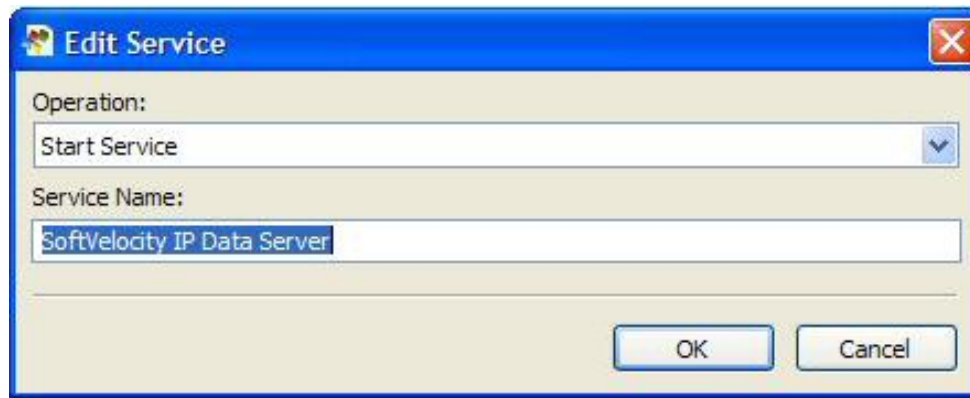


Figure 13. Starting the service again

I always put the "stop and uninstall" the service even in the main initial install because someone may be using it to overwrite a previous installation.

Error27 Fix

I asked SoftVelocity if Error27 had been fixed and they told me it is fixed in Clarion7. This was their reply:

"We're not sure when any 9059 AddOns will be released. However, when last tested, you could simply create the DataManager in C7 and leave the Client part in C6. The Error 27 issue was resolved on the datamanager side and this combination resolved the issue in our tests."

Conclusions

If you get an unexplained "invalid record structure" error when you *know* the file structure has been updated correctly, then you need to rebuild ipreq.cfg. I find the "ritual" specified here to be the way I get the most reliable results and I don't try to short-circuit it because it has cost me lots of days.

Chuck's method of stopping and starting the service during installation really works every time, and is quick, neat and invisible to the user.

Dermot Herron grew up and went to school in Rhodesia, and earned an electrical-engineering degree at the University of Cape Town. In the 1970s he worked for the Rhodesian Post Office, which was then also the telephone company. He was given the use of an HP9100, which was the very first programmable desktop computer, to help with the design of party-line telephones. He totally fell in love with computers then and there. Leaving the Post Office, he traveled for four years, worked in Canada (as a cabinet maker) and in England (as a microprocessor engineer) and then immigrated to New Zealand. But Dermot learned that once Africa gets into your blood you are doomed! Now he runs his own company in Johannesburg, writing software to send messages to mobile phones. He lives on a 19-acre plot with a river, and spends his off-time fixing things.

Reader Comments

Posted on Thursday, December 03, 2009 by Mark Riffey

Dermot,

A question about one of the comments you made...

"Double-click on the SessionTimeoutMinutes value (10, edit-in-place) and make it 10000 to prevent anything timing out. (bigger numbers don't store). Click OK."

That seems like potentially performance de-hancing advice. What am I missing by thinking that?

thanks,

Mark

Posted on Thursday, December 03, 2009 by Dermot Herron

Hello Mark,

I had trouble with the remote computers disconnecting and then complaining that they couldn't see the data. I put in the "keep connection alive" but I may have got it wrong and it still disconnected.

I use the IPDriver mostly on a LAN to protect my data from damage by power-fail at the remote computers so I don't EVER want automatic disconnection with a timeout. Disconnection then only occurs on shutdown or re-boot of the remote computers and that is fine. So 10000 minutes (nearly 7 days) was long enough for my purposes.

If one of the remotes "hangs" and everyone has gone home, I can always run IPSRVMGR and stop the service and re-start it and this guarantees to disconnect. (Incidentally, this also prevents remote computers connecting while I am working on the server. My bigger customers have 50+ connections and

someone always wants to come in...)

Dermot

[Add a comment](#)

Clarion Magazine

Guest Editorial: Is The Clarion Paradigm Dead?

by Robert Barton

Published 2009-11-23

Is the Clarion Paradigm dead?

First you must ask what you understand by the Clarion Paradigm.

The [SoftVelocity web site](#) gives me what I understand it to mean:

Clarion is a data-centric Advanced Rapid Application Development (ARAD) tool featuring roundtrip code generation that preserves all of your own hand-written code while allowing you to re-generate your application as often as needed. It provides reusable metadata to quickly create "corporate quality" applications to manage business data.

This is an excellent definition for my purposes, and also extremely comforting. I have never approached systems design from a computer science point of view — just business first and let's find a set of tools to create the business system and solve the business need.

Why do I need this? Productivity is the answer. The less work to create the system the better, provided that the system so created matches the business requirements in a scalable and reliable manner and is fit for purpose. Systems that take too much effort to produce are seldom timely, cost effective or fit for purpose.

So why Clarion?

When I started [Barton Hills Associates](#), the original Clarion for DOS product seemed to offer a means of attaining this goal. I obtained a copy and started out using it to build business systems. Prior to this, I had been involved with the traditional way of computing, i.e. a data processing department in a company. Too many layers, too much misunderstanding, too much paperwork, too little timely product — I could go on! It was not that something could be achieved — it usually was and often workable — but that so much more must be achievable with the right tools. But were the "right" tools available?

The Clarion Paradigm, to me as a Clarion Developer, is the combination of templates allied to the data dictionary and the application generator; it's a system that works in a very business-friendly way.

When I use the term "Clarion", I do not just mean the basic product but all the extra tools that are available under the Clarion umbrella, including third party products. It is a tribute to the basic soundness of the Clarion design that such extensions are available.

It seems to me that Clarion was the right tool! Productivity was assured. That was twenty years ago — so is it still the case today?

Clarion Progress

When Clarion progressed from a DOS product to Windows, it took some time before the business functionality caught up with what was available in the DOS product. Once that happened, the functionality started to improve in a major way with the capability of the add-on products giving a significant boost to the capability to build business systems.

The improved functionality was both in providing better solutions and in providing easier implementation and control of the business life cycle.

For example - [Capesoft's FM2](#) provided an effective means of controlling the changes in file systems that are an inevitable part of the life cycle, and [Lodestar's RPM](#) showed how to tackle the reporting required in a business system. Their simplicity of use meant that the I could focus on the business needs not the underlying code.

The above are just illustrations of what was available to move the business case along. Other products like Data Modeller, ReportWriter(both part of the main Clarion package), and [UltraTree](#) provided further evidence of the productivity available in the Clarion environment. When I look at the contents of my templates tool box I realise just how much functionality is available to create effective business systems, and how much I owe to the creators of these tools over the last twenty years.

In those days Clarion provided a superior solution to business design — in my opinion the best available!

Time does not stand still, and neither does the competition or the technology that is available, especially that thing called the Internet (and the "cloud")! I have to say the years between 1998 and 2006 were disappointing to say the least. In saying this I'm not commenting on the problems of Clarion's ownership etc. but on the growing gap in business functionality.

There were two reasons for this — the competition advanced at a pace that eroded the Clarion advantage and the Clarion product itself marked time. Notable exceptions were the introduction of [CapeSoft's NetTalk](#) and similar tools from [CHT](#), which provided the Clarion developer with a worthy set of tools to connect to the outside world.

Unfortunately these tools required a fairly high level of technical knowledge, so they were not always as productive as they could be from a business point of view.

At last Clarion is moving forward again, on both the Win32 and the .NET fronts. But does it still offer what I thought it did when I started using it twenty years ago? The difficulty in answering that is that my requirements (and more importantly my customer's requirements), have significantly changed over that period of time.

What is required now

So what has changed? Both the business requirements and the computer technology that is available.

Business expects far more from their systems today than they did fifteen years ago, when Clarion for Windows first appeared. We, business solution designers, need a higher level of abstraction to keep pace with business requirements. Building blocks need to be better, by offering more functionality than ever before without any increase in the effort to utilise them. An example of this is [ProScan](#) and [ProImage](#) products. These are not tools in the traditional sense but applications in their own right, yet they are part of the Clarion family. What they offer is a higher level of usability to the designer — full of capability yet simple to use. Document scanning — install ProScan — job done! OK it's not that easy, but it's not far away from that!

We need much more of this type of functionality but it's not there. For instance, the systems world has moved to SQL databases — how easy is this in Clarion? As easy as using standard TPS files? You have to be joking!

Computing must become simpler, not more complex to implement. Did the .NET world start out with that idea in mind? Perhaps. Certainly the idea of building blocks and creating functionality by assembling these building blocks is there, but as much as it is a computing technician's dream it is, at present, a business developer's nightmare.

The near (Clarion 7) future

What I would like to see happen in the very near future, in Clarion 7.x, is this:

Clarion (the base product) needs to stabilise and become really useful - maybe 7.1 is that product. There are substantial hooks within the IDE to move development tools forward. My especial hope is this area is to sort out SQL so that Clarion developers (and their customers) can use SQL databases painlessly.

On the web side, both Capesoft and CHT have special web software that nearly meets the bill, but is not yet developed enough to be truly regarded as a building block — more work needs to be done in this area so that the developer can make substantial applications using the available software (I'm well aware of the applications already produced). All these comments are not criticisms as such - more a plea for better tools! Better tools = better applications = better business for the Clarion Developer.

We already have suitable tools for setting up systems for install so that area is not a particular problem, but we do not need any more "help" from MS in this area — there is already too much interference from the OS provider. MS appear to be dedicated to complexity. I want simplicity which means in an MS environment we need better tools to remove this complexity from our development.

A welcome addition to the Clarion toolset is the [Noyantis](#) template family which has provided a simple method of using the up to date (.NET) visual paradigm for Windows. Other vendors like [Ingasoftplus](#) are providing Clarion wrappers around .NET controls. Clarion 7.1 contains a report writer from the .NET world — this time supported by SoftVelocity. At this stage between the Win32 and .NET world Clarion can still provide a good paradigm to produce effective business systems.

But we are gradually moving to a purely .NET world

The .NET future

The .NET world appears to be very class/code based — this is going in the wrong direction. It's not that there's anything wrong with object-oriented development; it's that there's too much code there to use and manage effectively. We do not need more code, we need more productivity. If the Clarion paradigm can support this by the use of templates etc. then perhaps we can once more gain a substantial development advantage. And this has to be the case, otherwise why would we continue to use Clarion — just for old time's sake? We must use it as a means of propelling our businesses into the future.

The .NET Clarion product is crucial to our future as it will hopefully unify more of the computing environments so that we can move business paradigms from one computing platform to another. I was listening to a Clarion Live Webinar the other day, where all the different .NET platforms were being mentioned — WinForms, WebForms, Compact etc. This horrifies me — I want simplicity not complexity — but what do I know, I'm only a customer trying to do a business job! But the pace of SoftVelocity is slow and the market is not waiting on them. There must be a way of speeding this up, otherwise the window of opportunity will no longer be there.

I reiterate, the tools we are using are good (perhaps better than that!) but not good enough for the future. We need tools that offer simplicity to install and use that have greater functionality than at the present. What the size of these tools should be is a good question — for example a complete ERP system would be overkill, but a bill of materials with data entry, browses, print out and calc. and file structure could well be very useful! How much of that is already available? Nearly all, at least as individual bits and pieces! But it needs to be available as a useful whole, a superset of what is already available. Rather in the same the way ProScan is built using ImageX and other Clarion template sets.

Will these new improved tools come from the Clarion world? I somehow doubt this, but would like to think so, having had a relationship with the best of the clarion tool providers over the last twenty years. Some will move with the .NET product, one would think, but would you bet your business on Clarion.NET becoming a successful product?

The .NET world really needs a product with something like the Clarion paradigm. Have SoftVelocity the ability to provide this? The building blocks are there, but they must change to fit the .NET world, if they want to be part of that future.

We cannot tolerate different sets of templates for the different types of .NET platforms — the paradigm must embrace this so that a unified application is possible (well maybe a few nods in a specific platform direction). By this I mean the templates used to create the business application, the unique business logic. What happens down stream from there is of small concern to the business developer. I am well aware that this is a considerable simplification but I am trying to get a point across.

Time is not on our side - too much of the first decade of the 21st Century has seen Clarion marking time. We need to progress swiftly to catch with some of the alternatives and more importantly to provide better business solutions so that we have a future within our own countries.

The task is daunting when expressed this way — do we have a track record from SoftVelocity that makes this achievable? SoftVelocity will have to change to deliver this. The [Australian DevCon](#), perhaps, gave pointers on how this can be achieved, but we are a long way to receiving some proof that it is going to happen — let alone a "gold" version of the product.

Clarion.NET is not going to be the first to market, but there is a long list of unknown products that were — but if you are not the first to market, when you arrive you had better have got it right! There will be no second chances unless you have the budget of a Microsoft!

Do I want to see a new Clarion rise to the task? YES! I would much prefer to use a Clarion solution but there are a myriad of examples in the market where the "best" product did not survive.

I hope this note can open up some debate to carry Clarion onward.

Apologies

I apologise for this being from a business systems point of view, but I am not contrite in any way that "coding" etc. has played a very little part in my arguments. Coding will still be required but must be minimised so as to reduce error and thus unreliable applications. Of course, the basis of any "product" in the computer world is code but that is for people who write code! We, Clarion Developers, are business system developers and use tools to out-produce our competition while producing business solutions for our Customers.

I use a large number of templates from add on suppliers that have not been mentioned here — there is no intention to praise or otherwise by their omission - just that they have not been mentioned to illustrate a point! Without them and Clarion, I would not have a business and certainly not been able to continue for over twenty years.

My comments may well be stating the situation in black and white terms - the reality is somewhere between. There are many helpful approaches by the add on developers that help minimise the technical aspects of their product. This is very welcome — I think CHT provides a good example in this area with their comprehensive set of demo applications and the Jump Start utility templates available for insertion direct into the developer's application.

There is a lot of what "I want ..." in this article, but I will point out that I have backed this over the years with considerable use of my wallet! This might just give me some rights in this matter.

Reader Comments

Posted on Monday, November 23, 2009 by Paul Blais

"Computing must become simpler, not more complex to implement."

I'll take a slight issue with this comment. The world is far more complex and the ideas and concepts are more complex. The only answer is more encapsulation. At the highest level, the complexity of business really is much more complex. Simple programs won't make the mark unless you confine yourself to installation.

We can not complain that programming needs to be more simple. It must meet the levels of complexity that exist. Totalling an invoice and printing a statement is no longer enough. It used to be almost the whole project with a big budget and a lot of hours. We need to perform rocket science on a daily basis.

If we can't get past low level SET / NEXT logic we won't get there. It's like writing assembly level code. When I started programming it was not uncommon.

.....
Posted on Tuesday, November 24, 2009 by Dave Harms

Paul,

Over the years Clarion programming has definitely become more complex, in response to the growing complexity of business software requirements.

In the early days Clarion appealed greatly to non-professional and beginning developers, folks who had a specific business need to solve and who used Clarion to accomplish that goal. I'd go so far as to say that was Clarion's primary market in the CPD days.

And as development needs became more complex, Clarion grew in power. But along the way quite a few of those CPD folks dropped off, and in some cases I think that's because programming was becoming more complicated and these folks weren't prepared (sometimes for quite valid reasons) to make the investment in new skills they would need for, say, object-oriented Windows development using Clarion.

This situation isn't unique to Clarion. I think every Clarion developer with an interest in .NET and the future of Clarion should listen to this Dot Net Rocks podcast:

<http://www.dotnetrocks.com/default.aspx?showNum=476>

What you'll hear is that exactly the same situation faces .NET developers. And there's a very real concern that the kinds of developers Clarion has always appealed to are exactly the developers being excluded from this brave new world by the overwhelming complexity (or so it seems) of .NET development.

The question, I think, is how will Clarion.NET address that complexity. And I'm not talking about Clarion#, the language - to me that's a non-issue. As Z himself has said, there's really not that much difference when you compare Clarion#, VB.NET and C# source for a given app.

It's all about code generation. How will the Clarion.NET AppGen bring about the same kinds of benefits we've enjoyed in the Win32 version? And it really is a more complex task because the Win32 version relies quite specifically on the Clarion window designer and a very limited set of custom controls. In .NET, the AppGen has to be able to work not just with a virtually unlimited control set, but also with a completely new environment, WPF, where the concept of a window designer and the architecture of applications is radically different than in WinForms.

SV is confident of their future - that's good, I'd be worried if they weren't. But as Robert points out, we really don't know yet what the new AppGen will look like. I'd add that we also don't know how it will address WPF development, which is already becoming a requirement for some developers.

So how about it, everyone? What kinds of features are you looking for in the new AppGen? Do you want to do exactly the same kind of development you've always done, just using an expanded set of controls and cranking out .NET code on the back end? Or do you have a vision of something quite different?

Posted on Tuesday, November 24, 2009 by Tom Drum

I have been a Clarion developer since 1991. A "developer", not a code writer? Thanks to the plethora of 3rd party tools out there, I haven't had to be a "coder". Nor have I ever wanted to be a "coder". Do I know Clarion coding? As much as I need to I was originally a COBOL coder and a Clipper coder (remember Clipper?). I didn't like being a coder. That's why I started using Clarion, way back then. And now? I'll be honest. I still do most of my embedded coding using the "old" structural language. Not OOP. Why? Cause I don't really understand OOP, at least not yet (remember, I WAS a COBOL coder). There's never been a pressing need for my to grasp OOP. Until now.

Why? Because I am not so sure about the future of SoftVelocity. I want to believe that they can come up with a product that I can use, but I'm starting to have my doubts. SoftVelocity develops Sloooooooooo. I work in a governmental "industry" that is demanding more and more services for the public. Now, everything must be browser based and on the internet/intranet. Browser based applications using Clarion? How? Well, I tried Clarion's internet templates. What a joke. The Application Broker running multiple instances of .exe files on the server was a nightmare. So, I tried Clarion/ASP. WAY to cumbersome (and yes, I do know HTML). Browser based? Hmmm. I finally decided that if I was going to be faced with creating browser based application (and keep my job), I was going to have to look at other development platforms.

So, far the "best" I have found is IronSpeed Designer (<http://www.ironspeed.com>). I've downloaded their free edition and have played around with it. It seems to me that it has the same business paradigm as Clarion. AND, it creates .NET code in either c+ or Visual Basic.NET. AND it is sudo-template based. BUT, that creates two problems for me. Learning a new language, and learning OOP. So I am now starting to. Everyone I've asking who knows both languages tells me that Visual Basic.NET is much easier to learn than c+, so that's the route I'm taking.

I don't want to have to start over. I HATE OOP! I LOVE Clarion! Yet, I can no longer develop the applications I am asked to using good old Clarion. And I don't see a .NET application generator anytime soon coming out of Pompano Beach. So what else can I do? I believe IronSpeed Designer is at least a start at getting me into the .NET world. I'll continue using good old Clarion 6.x to maintain applications I have produced (as long as I'm not forced to magically make them web based). But, I now realize (in my mind anyway) that Clarion is already too far behind in the race. I no longer have an advantage by developing in Clarion. In fact, developing in Clarion has become a disadvantage. I'm going to miss it. It's been like a loyal old dog to me. Always there for me (along with great 3rd party products) when I needed it. But, it's gotten old, no longer the fastest dog at the park. I think that it's time, for me at least, to put the old buddy to sleep. For it's own good and mine. Time to start looking for a new puppy. Sure puppies are a lot of work, but once they train you (which is usually the way it

is), you'll soon forget about your "old" buddy. And realize it was the right thing to do. That it truly was time to move on...

Posted on Tuesday, November 24, 2009 by kevin plummer

I've also been desperate to move my products to the internet. I have wasted a lot of time and money over the years trying Clarion add-on's and non-clarion solutions. For me and my clients web based solutions are the future.

I used GoGlobal to get my windows apps on the internet but it was only ever a temporary fix to buy me some time.

I was holding my breath waiting for Clarion7 .Net to save the day... well you can only hold your breath so long. Although I have subscribed, my enthusiasm has severely died.

I upgraded to an early build of Nettetalk 4. I had pretty much no knowledge of HTML, CSS or Javascript. It didn't seem to matter, Bruce's templates took care of everything. The early builds were basic and some pretty flaky. But it has matured into an amazing and stable product that is only restricted by your imagination.

So much so that 12 months ago I decided to replicate my main big app to the internet via Nettetalk. There is no conversion wizard so be prepared to redo all your GUI for the web - it is painstaking but no more painstaking than redoing it all again from scratch under windows.

The advantage I found with some R&D was that because it is all clarion code on the backend I didn't need to learn another language but better still I could drill into existing reports and functions - re-writing this stuff in another language I just wouldn't have done. It took me about 3 months to convert the app.

I'm yet to use Nettetalk 5 but what I have seen Bruce has added more features (like hooking in PHP pages) and using JQuery to handle AJAX and styling which according to his webcast opens up a lot more opportunities for the end developer than what we have available via the templates.

I still pretty much don't know much HTML or Javascript but to stylize my web app I have had to play with CSS to get the look and feel I wanted. I think NTW5/JQuery makes this easier than what it was - although it wasn't that hard with tools like FireBug.

I hope Bruce continues to develop this product with the vigor he has! I have banked my future on NT.

Kevin

Posted on Tuesday, November 24, 2009 by kevin plummer

I've also been desperate to move my products to the internet. I have wasted a lot of time and money over the years trying Clarion add-on's and non-clarion solutions. For me and my clients web based solutions are the future.

I used GoGlobal to get my windows apps on the internet but it was only ever a temporary fix to buy me some time.

I was holding my breath waiting for Clarion7 .Net to save the day... well you can only hold your breath so long. Although I have subscribed, my enthusiasm has severely died.

I upgraded to an early build of Nettetalk 4. I had pretty much no knowledge of HTML, CSS or Javascript. It didn't seem to matter, Bruce's templates took care of everything. The early builds were basic and some pretty flaky. But it has matured into an amazing and stable product that is only restricted by your imagination.

So much so that 12 months ago I decided to replicate my main big app to the internet via Nettetalk. There is no conversion wizard so be prepared to redo all your GUI for the web - it is painstaking but no more painstaking than redoing it all again from scratch under windows.

The advantage I found with some R&D was that because it is all clarion code on the backend I didn't need to learn another language but better still I could drill into existing reports and functions - re-writing this stuff in another language I just wouldn't have done. It took me about 3 months to convert the app.

I'm yet to use Nettetalk 5 but what I have seen Bruce has added more features (like hooking in PHP pages) and using JQuery to handle AJAX and styling which according to his webcast opens up a lot more opportunities for the end developer than what we have available via the templates.

I still pretty much don't know much HTML or Javascript but to stylize my web app I have had to play with CSS to get the look and feel I wanted. I think NTW5/JQuery makes this easier than what it was - although it wasn't that hard with tools like FireBug.

I hope Bruce continues to develop this product with the vigor he has! I have banked my future on NT.

Kevin

Posted on Wednesday, November 25, 2009 by Bruce Johnson

Hi Tom,

I'm biased, I know, so read this with the pinch of salt it deserves.

But if you want to see how to do real programs on the web, using Clarion code and templates, then check out

http://www.clarionlive.com/index.php?option=com_docman&task=cat_view&gid=42&Itemid=57

I recommend you watch Webinar #2, which was about creating web apps with NetTalk 4, and the follow up, webinar #35, which showed some of the new stuff in NetTalk 5.

I think you may find that it brings some of the web requirements you need, to the Clarion you already know. There are a lot of developers using it daily (like Kevin) and it continues to be an actively developed bridge from Clarion to web apps.

There's no Appbroker, or indeed anything else, required. And it doesn't follow the appbroker model, so it scales easily to thousands of users.

Cheers

Bruce

Posted on Wednesday, November 25, 2009 by Dave Harms

I don't have any vested interest in NetTalk, but it's probably the most important third party tool out there right now.

No, it isn't .NET. But I think it's the best way right now to preserve your Clarion investment while moving apps to the web. Anyone needing a web solution right now should be looking seriously at this product. CapeSoft's decision to use JQuery and ThemeRoller in NT5 means your web apps get a very contemporary look and feel.

And yes, I'll have a review/preview of NT5 in the mag, hopefully next month.

Dave

Posted on Thursday, November 26, 2009 by Rhys Daniell

I've aired my views at length here before, so I'll save you from the rant :-)

To me, the 'Clarion Paradigm' is to turn business rules into stable software in the shortest possible time.

CPD (DOS) absolutely got this, and created the core of the Clarion community but attracting people who were probably best described as business analysts rather than programmers.

Since then the developers of Clarion have lost their way a bit. In their defence, this is because two powerful forces have made software development massively more complex than it should be. These are

- tool developers, in particular Sun and Microsoft, who realise making programming more productive is going to harm their business, and
- the majority of programmers, who avidly chase new paradigms without any thought of productivity outcomes. I blame boredom with their toolset for this.

As I've said here before I think .Net is a massive mis-step for both those reasons.

However, with their forthcoming support for .Net, SV have a very real opportunity to turn things around. If they are successful in maintaining the 'Clarion Paradigm' while producing .Net applications then they will have a tiger by the tail.

I wish them well with this and I look forward to reaping the rewards. But I'm certainly not going to bet my business on it happening any time soon. What's more, I'm quite concerned that by the time it's out there and stable MS will have announced the *next* programming paradigm, and the thundering herd will be headed the other way again.

Posted on Thursday, November 26, 2009 by Dave Harms

Rhys,

Dunno about the next new *programming* paradigm (dynamic languages?) but the thundering .NET herd is already headed for the next new UI paradigm: WPF/Silverlight, a.k.a. XAML.

Dave

Posted on Thursday, November 26, 2009 by Rex Kersley

Quote:

"My especial hope is this area is to sort out SQL so that Clarion developers (and their customers) can use SQL databases painlessly."

According to recent ClarionMag surveys about 50% are using SQL. In my experience it is pretty painless.

Rex

Posted on Monday, November 30, 2009 by Majodi Ploegmakers

Clarion and SQL, no problem, great match, we're doing that for over 12 years now.

Web apps: NetTalk. We use it successfully, not much coding, great result, fast, winner for the time being for the not so big solutions where you need to be pragmatic to be able to earn a buck.

.NET - that's where it hurts. Do we have to go into that direction? I'm afraid it's unavoidable. Without a Clarion-like tool for .NET most of us "business to solution translators" are fried. Using mainstream .NET tools (VS) is back to primitive spartan coding.

Can we expect SV to save us? They know what we are looking for. They have knowledge on how to achieve it. Do they have the resources to actual make it happen? I doubt (seeing their achievements over the past years).

And of course there are these many choices you are faced with in the .NET world. Choices that are moving into new directions almost overnight. They would need to embrace the right (current) design patterns and technology and keep up with it as good as possible. Currently for desktop apps that would mean (for me) MVVM with WPF. But at the moment sharpdevelop doesn't offer a xaml designer so they have to opt for winforms. Is that bad? Not per se but will they be able to "upgrade" to WPF in a timely manner.

And what architectual pattern will they choose? If not an MVVM alike variant will it be separated into multiple tiers or be 90's mixed UI/business logic kind of code.

I've watched the same delemma early to mid 90's when they had to make a shift from DOS to Windows. BB was still in charge. A midway solution, a graphical DOS CDD was the search path to the real solution CW. To get there, good people were hired or otherwise added by a merger. These things do need money and energy. Will they pull it off this time? Ehh.. I hope, but at the mean time I look at initiatives like GSF, Genwise Software Factory, <http://www.genwise.nl>.

We particepated in their GenWise Studio product beta testing a bit. Then they saw that they needed a different approach of integrating with Visual Studio and utilizing DSL's with WPF. All very promising. They will be releasing a beta soon now.

Is that the new holy grail? Not quite yet I guess but promising. At the moment right choices of architectual patterns and technology but still far away from code generation to a level to be comfortable with (still too much spartan coding to do). It just needs to grow. For .NET brains they are doing magic already but for us spoiled it is not mature yet.

In the mean time; wait, learn, use what is out there.

Posted on Monday, November 30, 2009 by Dave Harms

Thanks for your thoughts, Majodi. I think it's worth adding that the GenWise folks don't see themselves as a competitor to Clarion. Although GenWise does generate code, it doesn't have anything quite like the AppGen, and it really is more of a hand coder's environment where a certain amount of the grunt work is done for you.

Dave

Posted on Friday, December 04, 2009 by Rhys Daniell

Just to clarify my SQL comment - I think in general Clarion and SQL get along pretty well, with a few simple techniques you can make the most of SQL without any significant limits.

However doing any serious coding in SQL, e.g. multiple big stored procedures that call multiple functions is a PITA, I'd like to see a Clarion-type IDE which displays code relationships and makes it easy to drill down to the bits you want to work on. Heck, I'd like to be able to edit SQL in Clarion!

[Add a comment](#)

Clarion Magazine

The Problem With Embeds, Part 3: The Invoice App

by Dave Harms

Published 2009-11-26

This is the third article in an ongoing series on Clarion embed points and how almost all Clarion developers use them in the wrong way. If you haven't read the [first article in the series](#), then you might find that statement offensive. You might be thinking "What does Harms know anyway? I'm a professional developer. I *know* how to write embed code."

If that's you, stand down from red alert. First, I include myself among the offenders. I've written embed code for years, and I've almost always done it wrong, even when the code itself was pretty dang good. Please, just go back to [Part 1](#), read it, and *then* tell me where I'm off base. (Don't bother with [Part 2](#) just yet, unless you're really keen. Part 2 should probably have come later in the series. Oh well.)

Once again, here's the idea in a nutshell:

Embed code should *not* contain business logic.

Put another way, your app contains code that gives it unique value to your users/customers. *Don't* bury that code inside embed points. You can *call* it from embed points (you'll almost certainly have to) but don't let it *live* there. Good code needs a home, a place where it can be conceived, nurtured, and yes, tested and disciplined.

So what does that home look like?

I think it looks like a class library, and I've taken some heat for saying that. In this article I'll defend my point of view, and I'll do it by beginning a transformation of the Invoice application which ships with Clarion.

The Invoice app

The example Invoice app isn't something you'd want to use in real life, but I've chosen it because we all have it, and because many of us have, at one time or another, had to deal with invoicing.

Figure 1 shows the customer order window, which has a child browse showing line items for each invoice.

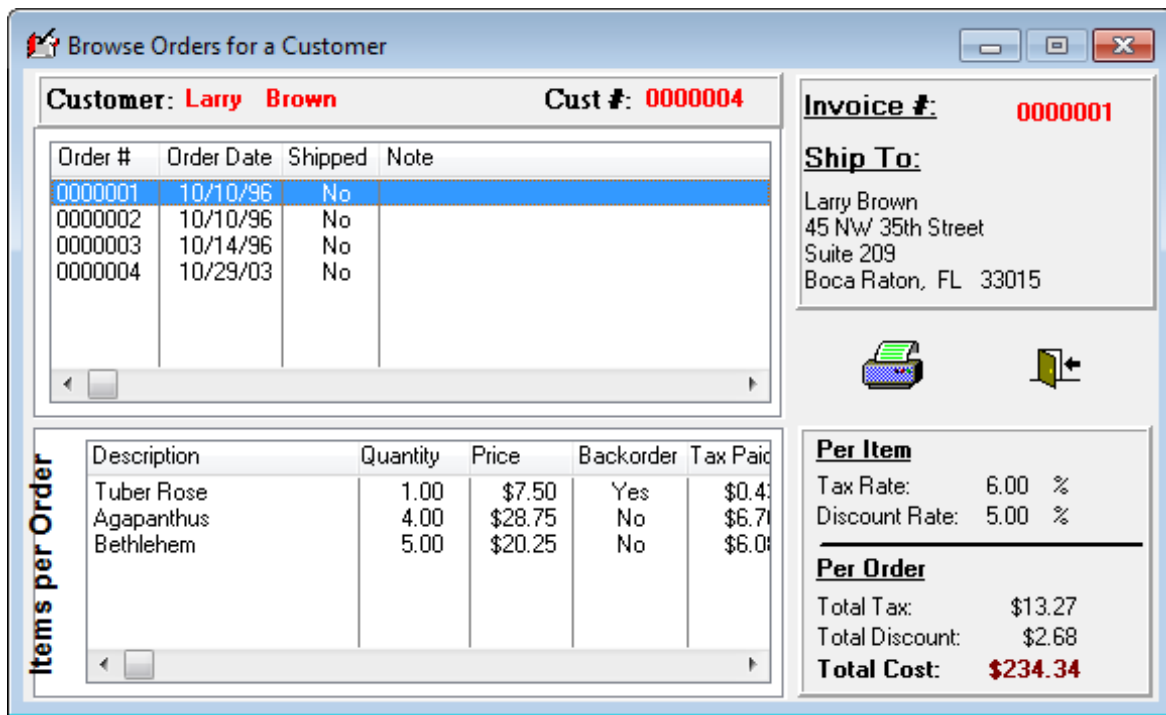


Figure 1. Browsing orders for a customer

Figure 2 shows the order detail form. Yeah, I know, none of this is pretty. But that's okay - it's old example code, and Clarion devs sometimes have to deal with old code.

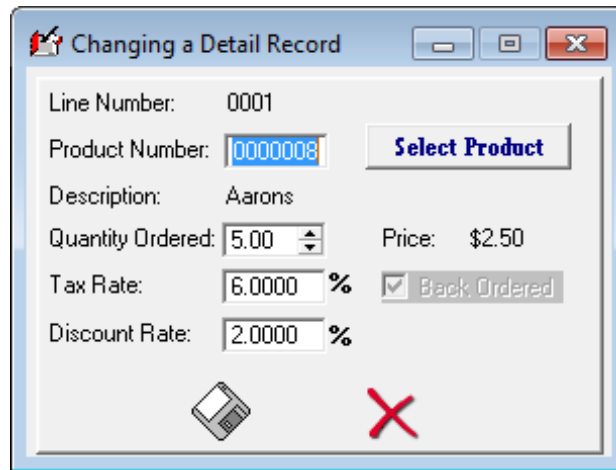


Figure 2. Changing a detail record

Here's the Detail file structure:

```

Detail      FILE,DRIVER('TOPSPEED'),PRE(DTL),CREATE,BINDABLE,THREAD
KeyDetails  KEY(DTL:CustNumber,DTL:OrderNumber,DTL:LineNumber),NOCASE,OPT,PRIMARY
Record      RECORD,PRE()
CustNumber  LONG
OrderNumber LONG
LineNumber  SHORT
    
```

```

ProductNumber      LONG
QuantityOrdered    DECIMAL(7,2)
BackOrdered        BYTE
Price              DECIMAL(7,2)
TaxRate            DECIMAL(6,4)
TaxPaid            DECIMAL(7,2)
DiscountRate       DECIMAL(6,4)
Discount           DECIMAL(7,2)
Savings            DECIMAL(7,2)
TotalCost          DECIMAL(10,2)

```

```

      END

```

```

    END

```

There's a fair bit of embed code in the UpdateDetail form. The part I'll tackle first is the code that calculates totals and taxes:

```

CalcValues ROUTINE

```

```

  IF DTL:TaxRate = 0 THEN

```

```

    IF DTL:DiscountRate = 0 THEN

```

```

      DTL:TotalCost = DTL:Price * DTL:QuantityOrdered

```

```

    ELSE

```

```

      LOC:RegTotalPrice = DTL:Price * DTL:QuantityOrdered

```

```

      DTL:Discount = LOC:RegTotalPrice * DTL:DiscountRate / 100

```

```

      DTL:TotalCost = LOC:RegTotalPrice - DTL:Discount

```

```

      DTL:Savings = LOC:RegTotalPrice - DTL:TotalCost

```

```

    END

```

```

  ELSE

```

```

    IF DTL:DiscountRate = 0 THEN

```

```

      LOC:RegTotalPrice = DTL:Price * DTL:QuantityOrdered

```

```

      DTL:TaxPaid = LOC:RegTotalPrice * DTL:TaxRate / 100

```

```

      DTL:TotalCost = LOC:RegTotalPrice + DTL:TaxPaid

```

```

    ELSE

```

```

      LOC:RegTotalPrice = DTL:Price * DTL:QuantityOrdered

```

```

      DTL:Discount = LOC:RegTotalPrice * DTL:DiscountRate / 100

```

```

      LOC:DiscTotalPrice = LOC:RegTotalPrice - DTL:Discount

```

```

      DTL:TaxPaid = LOC:DiscTotalPrice * DTL:TaxRate / 100

```

```

      DTL:TotalCost = LOC:DiscTotalPrice + DTL:TaxPaid

```

```

      DTL:Savings = LOC:RegTotalPrice - DTL:TotalCost

```

```

    END

```

```

  END

```

CalcValues is called whenever the OK button is clicked, which means whenever the record is saved, on an insert or an update.

And I don't know about you, but I see one big bug staring me in the face already.

This routine updates values in a record, so that's hard data going into the database. If I retrieve a record from the database which previously had DTL:Discount and DTL:Savings values set, and I set DTL:TaxRate to zero and DTL:DiscountRate to zero, what will happen when I call CalcValues? This one line of code executes:

```
DTL:TotalCost = DTL:Price * DTL:QuantityOrdered
```

It appears there's no change at all to the discount and savings values.

But how can I test this? A few options come to mind. I could put in (shudder) Stop statements or Message statements. Or I could use [OutputDebugString and DebugView](#). Or perhaps I could just add the rest of the fields to the screen so I can see what's actually happening to the data.

In Figure 3 I've added the calculated fields as display strings. I've also added a button to force a call to the CalcValues routine so I can see the results of several changes at one time.

Line Number:	0004				
Product Number:	0000019	...			
Description:	Aster		<input type="checkbox"/> Back Ordered		
Quantity Ordered:	2.00		Price:	\$15.00	
Tax Rate:	0.0000 %		Tax Paid:	\$0.00	
Discount Rate:	0.0000 %		Discount:	\$0.00	
			Total Cost:	\$30.00	
			Savings:	\$0.00	

Figure 3. Displaying the calculated fields

In Figure 3 I've added a new detail line, and I've selected an Aster, quantity two. I've pressed the Recalculate Values button. You can see that the values all appear to be correct.

In Figure 4 I've changed the tax rate to 5%.

The screenshot shows a dialog box titled "Adding a Detail Record" with the following fields and values:

Line Number:	0004		
Product Number:	0000019	...	
Description:	Aster	<input type="checkbox"/> Back Ordered	
Quantity Ordered:	2.00	Price:	\$15.00
Tax Rate:	5.0000 %	Tax Paid:	\$1.50
Discount Rate:	0.0000 %	Discount:	\$0.00
		Total Cost:	\$31.50
		Savings:	\$0.00

At the bottom, there is a "Recalculate Values" button, a floppy disk icon, and a red "X" icon.

Figure 4. Setting the tax rate

So far, so good.

In Figure 5 I've added a 10% discount.

The screenshot shows the same dialog box as Figure 4, but with the Tax Rate changed to 10.0000 %:

Line Number:	0004		
Product Number:	0000019	...	
Description:	Aster	<input type="checkbox"/> Back Ordered	
Quantity Ordered:	2.00	Price:	\$15.00
Tax Rate:	10.0000 %	Tax Paid:	\$3.00
Discount Rate:	0.0000 %	Discount:	\$0.00
		Total Cost:	\$33.00
		Savings:	\$0.00

The "Recalculate Values" button, floppy disk icon, and red "X" icon are also present.

Figure 5. Adding a discount

Whoops. The discount is calculated as \$3.00, which is correct, at least for the total if not for the per-item price (the Invoice app has no formal notion of "extended" price). Why then is the savings only \$1.65?

In Figure 6, I've set tax rate to 5% and the discount to 1%.

Adding a Detail Record

Line Number: 0004

Product Number: 0000019 ...

Description: Aster Back Ordered

Quantity Ordered: 2.00

Tax Rate: 5.0000 %

Discount Rate: 1.0000 %

Price: \$15.00

Tax Paid: \$1.49

Discount: \$0.30

Total Cost: \$31.19

Savings: \$-1.19

Recalculate Values



 

Figure 6. A 1% discount and a negative savings

According to Figure 6, offering a 1% discount to a customer results in a *negative* savings. Interesting.

In Figure 7, the wheels have well and truly come off.

Adding a Detail Record

Line Number: 0004

Product Number: 0000019 ...

Description: Aster Back Ordered

Quantity Ordered: 2.00

Tax Rate: 0.0000 %

Discount Rate: 0.0000 %

Price: \$15.00

Tax Paid: \$1.49

Discount: \$0.30

Total Cost: \$30.00

Savings: \$-1.19

Recalculate Values



 

Figure 7. Setting the tax rate to zero

I've set the discount rate and the tax rate to zero, but I'm still showing a tax paid of \$1.50, a discount of 30 cents, and a savings of \$-1.19.

Wonderful.

Of course you'd never write code that's this buggy, and I shouldn't be too hard on the author of the app either since it's

nothing more than a demo.

We've all written code *like* this, with less-obvious bugs. Sometimes it's a long, long time before you even find out the bug is there. And what happens when you need to reuse the code somewhere else, say in a web version of your invoicing application (built with Capesoft's snazzy NetTalk 5, perhaps)? You make a copy, and guess what, you've just made a copy of an obscure bug. Let's hope you remember to fix the problem in both places when it finally comes to light.

But how did these bugs happen in the first place? Because Clarion's embed system, as powerful as it is, encourages untestable business logic that's too tightly bound to the user interface. It's easy to write code that's hard to test.

The solution is to take the code out of the embed and put it somewhere it can be both reused *and* tested.

The beautiful black box

The important thing about business logic is that you should be able to use it without having to know exactly how it works. That may sound a bit counterintuitive - after all, if it's your own code, shouldn't you know how it works? If you wrote it, yes, you should. But we all use black box code all the time. How many Clarion devs really know what's going on in Windows when a mouse is moved or a button clicked? How about the file driver system? Or even ABC? We don't need to know how these things all work; we just need to know how to use them.

You can take the same approach to your own software development. You don't need to remember how every bit of your application works (if you're like me, you *can't* remember all that stuff anyway). You just need to do a good job writing it in the first place, and you also need to write good tests for your code so you have confidence it's still doing what you originally wrote it to do.

Embed code is incredibly tedious to test, and almost impossible to test thoroughly, because you typically have to run the application, do a bunch of stuff on screen, and look at the results to see if the code works. That's the *only* way to test the CalcValues routine as long as it stays in the embed point.

But what's the first thing that happens if you take the routine out of an embed point and put it into a standalone source file? It won't compile. That's because it has references to app-specific data: the fields in the Details record. The code is entangled with the database definition, and that's not a good thing for either testing or re-use. Same goes for the user interface: if your code requires a certain window or control to be in existence, you're going to have a lot of difficulty with testing.

When it comes to black boxes containing business logic, you really don't want anything that's specific to one application, its database, or its user interface. All you really want to do is feed some data into that box and get back some other data.

In the case of the CalcValues routine code, you want to feed in:

- Unit price
- Quantity
- Discount rate
- Tax rate

and you want to get back

- Total price
- Tax amount

- Discount amount
- Total cost
- Savings

How to design a black box

Once you know the inputs and outputs, it's easy to design your black box. You just need to decide whether to create your black box as a procedure or a class.

Here's how you might declare a black box procedure to replace the CalcValues routine:

```
CalculateInvoiceDetailValues  procedure(*decimal price,|
                             long quantity,|
                             *decimal discountRate,|
                             *decimal taxRate,|
                             *decimal taxAmount,|
                             *decimal discountAmount,|
                             *decimal totalCost,|
                             *decimal savings)
```

Note that the procedure doesn't have any dependency on the Detail record, or on anything else that's specific to the Invoice app.

To use the procedure in the Invoice app you'd simply call it like this:

```
CalculateInvoiceDetailValues(DTL:Price,|
                             DTL:QuantityOrdered,|
                             DTL:DiscountRate,|
                             DTL:TaxRate,|
                             DTL:TaxPaid,|
                             DTL:Discount,|
                             DTL:TotalCost,|
                             DTL:Savings)
```

And here's how you could declare a similar black box class to replace the CalcValues routine:

```
InvoiceDetail  Class
Init           Procedure(*decimal price,|
                        long quantity,|
                        *decimal discountRate, |
                        *decimal taxRate)

GetTotalCost  procedure(*decimal totalCost)
GetTaxAmount  procedure(*decimal taxAmount)
GetDiscountAmount  procedure(*decimal discountAmount)
```

```

GetSavings      procedure(*decimal savings)
                End

```

To use the class you'd write something like this:

```

InvoiceDetail.Init(DTL:Price,|
                  DTL:QuantityOrdered,|
                  DTL:DiscountRate,|
                  DTL:TaxRate)

InvoiceDetail.GetTotalCost(DTL:TotalCost)
InvoiceDetail.GetTaxAmount(DTL:TaxPaid)
InvoiceDetail.GetDiscountAmount(DTL:Discount)
InvoiceDetail.GetSavings(DTL:Savings)

```

Notice that in both cases decimal values are passed by address, which is a Clarion requirement.

There isn't all that much difference between the two approaches so far. The procedural syntax takes a little less typing, but it's also slightly more difficult to understand. It's easier to make a mistake with eight parameters than it is with four, and if you wanted to be very sure the class was used correctly you could always create individual setter methods for each of the four initialization parameters (and report an error if not all were set before a value was requested).

The real problem with the procedural approach, however, is its inflexibility. In its current configuration you *must* pass all eight parameters, even if you only want one value back. You could make the parameters omissible, but then you have to make sure you omit the right parameters, which can also be tricky.

As well, the procedural code can really only be used *as* procedural code. That is, you're using the class pretty much as a set of procedures, but you can also use the class instance *itself*. You could, if you wish, create an InvoiceHeader object that contains a list of InvoiceDetail objects, such that you could model the entire invoice in memory. It's very difficult to do anything like that with procedural code. I'll get more into the advanced uses of the InvoiceDetail class in future articles.

It's also easier to extend the InvoiceDetail class without breaking existing functionality. As written, the CalcValues routine calculates total cost, including taxes, but not extended price excluding taxes. Adding a method to get the extended price would have zero effect on code already using the class, whereas with the procedural solution you'd have to make the additional parameter omissible.

In this very limited set of calculations and results the procedural approach is already starting to show some strain. What happens when things get really interesting, as when you need to start tracking multiple taxes instead of a single tax?

In other words, the evolution of programming from procedural code to object-oriented code isn't without very good reason. OOP really does give you a greatly expanded toolbox for solving programming problems.

Testing

Whichever approach you take, procedural or class, your code needs to be testable. And I don't mean the kind of testing that involves someone running an app, entering values, and making sure the results are correct. Yes, you do still need to do

that, but by the time your code gets to that point you should already have a high level of confidence.

The kind of testing I'm talking about is called "unit testing". The "unit" refers to the smallest testable parts of the application. In the case of the procedural version of the above code, it would involve testing a single call to the `CalculateInvoiceDetailValues` procedure. In the case of the class version, you would have a test for each of the methods which return a value.

That brings up another advantage of classes - it's easier to break testing down into smaller units.

The *idea* behind unit testing is that if all of the individual objects (or procedures) that make up your application work properly, you're much more likely to have an application that works properly as a whole.

As well, unit testing is a great way to prevent the introduction of new bugs as you make modifications to existing code. If you have a large suite of unit tests, and you make a change to some core code that affects how other methods or classes (or procedures) work, any bugs introduced by those changes stand a good chance of showing up in the unit tests.

One of the keys to successful unit testing is to automate the process as much as possible. You don't want to have to manually launch each test; instead, you want a way to run a whole bunch of tests and see the results. It's also advantageous to run all the unit tests as part of every build, so that bugs can be found as quickly as possible.

Another key is to avoid outside dependencies in unit tests, such as a requirement to use an actual database or a user interface widget. You want your unit tests to run as quickly as possible, and with little to no chance of breakage due to something on the outside of the tested code not being set up correctly.

In the case of the `CalcValues` routine code, isolation is pretty easy to achieve. Although the routine code operates directly on the fields in the `Detail` record, this isn't a requirement. The logic can all be made internal to the class; that way the class can be used anywhere at all without modification, and the only embed code needed is whatever it takes to assign the values to and from the class. Presto! You've successfully removed the business logic from the embed point!

What's next

There are a couple of ways to approach unit testing. One is to take existing code, refactor it if necessary to make it testable, and then write a bunch of tests to see whether it's working as you expect. The other is to start by writing tests, and then create the code to make those tests compile and then pass. This latter approach is called test-driven development (TDD) and can be enormously useful. Although I do have some existing code to work with, it's broken enough that I'm going to start fresh, using the routine code only as a reference.

In the next article I'll walk through the process of creating and running unit tests in Clarion, and then I'll flesh out the `InvoiceDetail` class and provide a refresher on some of the basics of object-oriented programming.

The downloadable source zip contains the modified Invoice app shown above, in C7 format.

[Download the source](#)

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

Reader Comments

[Add a comment](#)

Clarion Magazine

ClarionFAQ Update

Published 2009-11-30

Last month we launched a new Frequently Asked Questions site, [ClarionFAQ](#). If you've ever used the wildly popular [StackOverflow](#) site (for non-Clarion programming questions) our new FAQ site will look somewhat familiar. That's because we're using StackExchange to run ClarionFAQ.

Clarion developers are showing a lot of interest in [ClarionFAQ](#). Here's a list of recent questions:

- [What keyboard shortcuts are available in C7?](#)
- [How can I minimize a Clarion app to the system tray?](#)
- [What's the URL for the PTSS system for C7?](#)
- [control{prop use} = dynamicfile.column ? How?](#)
- [Queue Structure Generator](#)
- [Finger Print Scanner - \[DigitalPersona One Touch for Windows\]](#)
- [Please remember to vote!](#)
- [Why can't I have more than about 125 connections to my PostgreSQL server?](#)
- [How Can I Get A Global Template To Add Code To Every Procedure?](#)
- [Adding buttons & images to the Frame](#)
- [How do I run Clarion 6 on Windows 7 64](#)
- [How to print bills having diff page length on role printer using clarion one below other](#)
- [Clarion and API](#)
- [Capturing images from a webcam](#)
- [Error 33 - Record not available in Process \(SQL\)](#)
- [Print downwards then across](#)
- [How do you apply the same wallpaper to all of your windows?](#)
- [Printing on custom](#)
- [Calculate Nmbr of Years \(AGE\)](#)
- [Link Problem on app that used to work](#)
- [Date fields in SQL](#)
- [How do you install the Topspeed ODBC driver on a workstation](#)
- [RTF issues: inserting RTF from one control to another](#)
- [How can a global template share its settings across multiple APPs?](#)

- Using Das tag on condition to print
- Can a window be made to recalculate constantly, as the user is editing fields?
- Base 64 decoder ?
- Base 64 Encoder ?
- How do I add customized hotkeys C7/Clarion.NET IDE?
- How To Create New Odbc Connection If Current Connection Is Gone Before Firing Query
- How do you create a sub menu in an ABC browse right click popup
- When do you use TakeCompleted
- What kinds of problems do Clarion apps have with multicore CPUs?
- Is there a list of TPS error codes?
- How do I make a variable in a global dll template available to other dlls ?
- The "No Sheet" setting on a SHEET causes strange tab positioning in C6. Is there a fix?
- What kinds of issues (if any) do Clarion apps have with Windows 7?
- Do I need a commercial license to use MySQL with my Clarion application?
- Can I delete one table from a Clarion Superfile?
- Why Can't I View My Clarion 7 Encrypted Files In TopScan?
- What changes are there to the template language and AppGen in Clarion.NET?
- Where did you find the Cla\$Start details
- How do I convert a C7 application back to C6?
- How do you execute a procedure by address?
- How do I START a procedure by its name?
- Where can I find official announcements about Clarion 7?

Check out [ClarionFAQ](#), post your questions and answer questions. This is a free service offered by Clarion Magazine.

Reader Comments

[Add a comment](#)

Clarion Magazine

Have Your Say On All Things Clarion

Published 2009-11-30

Have you ever wanted to have our say about which features should be in Clarion, or what SoftVelocity's priorities ought to be, or have you just wanted to make suggestions that other developers could read and possibly support?

Now you can, at Clarion.UserVoice.com.

Clarion.UserVoice.com (a.k.a. CUV) is not part of or endorsed by SoftVelocity. It's the brainchild of business/marketing guru [Mark Riffey](#), and is one of the thousands of customer service sites provided by UserVoice.com, which provides tools to help companies listen to their customers. Mark has agreed to hand over the reins of CUV to Clarion Magazine.

So what can you do at CUV? Three things, really.

1. Post suggestions. Is there something you think should be part of Clarion? Is there some future direction you'd like to see the product take? Is there an issue that needs to be fixed?
2. Vote on suggestions.
3. Comment on suggestions

Here are some of the suggestions currently posted at CUV:

- Easy way to handle COM objects
- IDE remembers state of the embed tree
- C7's optional use of owner-drawn menus
- Full Microsoft "certified for" support (no more old w95/98 API)
- Deliver Clarion.Net Gold
- A real debugger
- Implement Clarion# Appgen inside Visual Studio

CUV is now a project of Clarion Magazine. While we can't absolutely guarantee that SoftVelocity will pay attention to what's posted on the site, we certainly will. And feel free to make suggestions about Clarion Magazine too!

Visit Clarion.UserVoice.com.

Reader Comments

Posted on Wednesday, December 02, 2009 by Kevin Erskine

UGhhhhh - Yet another website not endorsed by SoftVelocity talking about their products with little hope of them listening or even reading the site. I thought the purpose of the NGs where to 1) be endorsed by SV and 2) to be reviewed by SV 3) etc....

Can you sense the frustration level here :)

Kevin

Posted on Monday, December 07, 2009 by Paul MacFarlane

The nice thing about uservice is the ranking and voting. Not been done by other Clarion sites.

I've already started an account for one of my products.

[Add a comment](#)