Clarion Magazine

Clarion News

- » One Thousand Reasons to Subscribe to ClarionMag
- » Clarion 6.3 9054 In Compatibility Testing
- » SetupBuilder for Clarion Anniversary Special
- » Organize365 Price Increase
- » SetupBuilder Anniversary Sale
- » Free Utility: BackItUp
- » UK Jobs Site Updated
- » Clarion and the War On Terror (Update)
- » Alexander Ivanovsky Back Sept 10
- » Free ClaColor Utility
- » New Prices At ClarionWebHosting
- » Template Writing Course
- » Professional Icons $29.95 In August
- » Ace/Roma Icons Customers Upgrade Notice
- » Free PDF-Xchange PRO for Clarion Developers
- » GTL6.23

[More news]

Latest Free Content

- » SV Releases Clarion Roadmap
- » OSCON: Query Interfaces, PHP/Web 2.0, and TimeTravel Tables
- » OSCON: solidDB Beefs Up MySQL OLTP Choices
- » OSCON: Wednesday Keynotes
- » Clarion Tips & Techniques, Volume 3 Available Now

[More free articles]

Clarion Sites

- » RADFusion
- » PINVOKE.NET
- » SealSoft Company
- » Lodestar Software (CWaddons)
- » The Clarion Connection
- » BoxSoft (a.k.a. Mike Hanson)
- » CapeSoft
- » SoftVelocity Online Store
- » SoftVelocity Community Site
- » SoftVelocity Company Site

[More sites]

Podcast

SoftVelocity president Bob Zaunere is back on Planet Clarion. Andrew and Dave talk with Bob about the ConDev AVIs and what they really mean, what's involved in creating the new IDE, and the differences between Win32 and .NET screen formatters. 47:00:00, 16913K

Listen now

[Track lists, more podcasts]

Clarion Blogs

- » DevDawn

- » John Griffiths
- » Knobblegrud
- » JohnMo
- » SoftVelocity
- » Lesley Dean
- » DevDawn
- » Gary James
- » Clarion Magazine

Latest Subscriber Content

Our DevCon or No DevCon Sale!

September is traditionally the time for a Clarion DevCon and a ClarionMag sale. There's no DevCon this year but we're not letting that minor technicality stop us. Until Sept 15 new subscriptions are just $109 including all the back issues; renewals are $79. And we have weekly specials for subscribers! This week save $14 off the list price of any e-book. Next week: book deals! (Subscribe now and you can still take advantage of next week's special.)

Posted Tuesday, September 05, 2006

An Economical Record Status Control

Nardus Swanevelder condenses six controls and their prompts into a single line control displaying "user created/changed" status information.

Posted Monday, August 28, 2006

Commentary on UKCUG, Clarion Roadmap

Dave Harms comments on some additional screen shots from the UKCUG meeting, and SoftVelocity's newly published roadmap for Clarion.

Posted Monday, August 21, 2006

SV Releases Clarion Roadmap

Bob Zaunere has released SoftVelocity's roadmap for Clarion through 2007, with details on the Win32/Win64 versions and Clarion.NET for .NET 2.0 and 3.0.

Posted Thursday, August 10, 2006

OSCON: PC History, Tech Trends, and Wrap-up

This closing OSCON report includes Dave Bradley on the history of the IBM PC,

O'Reilly's take on technology trends, the riotous Damian Conway, and the conference wrap-up.
Posted Wednesday, August 09, 2006

OSCON: Keynotes, Database Tuning, and .NET Tools

OSCON coverage continues with a keynote address by Robert Lefkowitz, Google's open source project hosting announcement, PostgreSQL tuning tips, and a length list of .NET open source development tools.
Posted Tuesday, August 08, 2006

UK CUG Report on C7/Clarion.NET

Richard Rose reports on the UK CUG meeting, including news on Clarion 7, Clarion.NET, and the new IDE.
Posted Saturday, August 05, 2006

OSCON: Query Interfaces, PHP/Web 2.0, and TimeTravel Tables

Session reports include a progressively filtering search interface, PHP creator Rasmus Lerdorf on Web 2.0, and a nifty way to get snapshots of your database at any point in time.
Posted Saturday, July 29, 2006

OSCON: solidDB Beefs Up MySQL OLTP Choices

Seeking an opportunity to enter the open source marketplace, high end database provider Solid Information Technology has stepped up with a version of its OLTP database for MySQL, released in beta on July 25.
Posted Friday, July 28, 2006

OSCON: Wednesday Keynotes

OSCON's Wednesday keynote address featured Tim O'Reilly, Scot Yara, Anil Dash, and Mike Olson.
Posted Thursday, July 27, 2006

ClarionMag at OSCON 2006

Clarion Magazine is at O'Reilly's Open Source Convention this week, following the database track. Dave Harms sets the stage.
Posted Wednesday, July 26, 2006

Marking Time 3: Inter-Date Computations

Basic date math, such as adding days to a date, or finding the number of days between two dates, is easy in Clarion. Time math gets a little trickier, because of the way Clarion

assigns time values. And things really start to get interesting when calculations involve midnight rollover, and other operations involving date/time values on different dates.
Posted Thursday, July 20, 2006

Improving On The Non-Related Lookup Template

Clarion ships with a template that allows you to do a lookup from a table that is not related to the primary table. While very useful, it offers room for improvement, as Nardus Swanevelder demonstrates.
Posted Monday, July 17, 2006

Beautifying Clarion Applications

Beauty may only be GUI deep, but that's no reason to hide your beautiful code with an ugly shell. Roel Abspoel shows how to dress up your apps with that "glass panel" look.
Posted Thursday, July 13, 2006

Another Single Browse For Multiple Lookups

Several years ago Clarion Magazine published an article on using a single browse for multiple lookups. Nardus Swanevelder looks at the subject again, and comes up with some improvements.
Posted Monday, July 10, 2006

A Class Wrapper for the SimpleOCR API

While developing a document management system, John Dunn faced a requirement for scanned documents. To reduce data entry time and increase the accuracy of the indexed data he decided to add Optical Character Recognition (OCR) capabilities to the program.
Posted Thursday, July 06, 2006

Clarion Tips & Techniques, Volume 3 Available Now

Clarion Tips & Techniques, Volume 3, is now in print!
This is the biggest (at 700 pages) and best Clarion Tips book yet!
Topics include: database issues such as business rules, triggers, conversions and arrays; writing/modifying templates; C6 threading; handling browses, forms and reports; using Clarion with .NET; calling the Windows API; and much, much more!
Now for sale in the Clarion Magazine store for $79.95 (if you have a current subscription your price is $69.95).

Posted Tuesday, July 04, 2006

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

Printed Books & E-Books

E-Books

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

Printed Books

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

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

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

From The Publisher

About Clarion Magazine

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

Subscriptions

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

Satisfaction Guaranteed

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

Dave Harms

ISSN

Clarion Magazine's ISSN

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

About ISSN

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

Clarion Magazine

Clarion News

Search the news archive

One Thousand Reasons to Subscribe to ClarionMag

Check out these alphabetical listings of free and subscriber-only articles published in Clarion Magazine.
Posted Tuesday, September 05, 2006

Clarion 6.3 9054 In Compatibility Testing

SoftVelocity has released build 9054 to third party product vendors for compatibility testing. Projected date for the public release is the middle of next week. There are a quite a few important fixes, as well as some nice new features.
Posted Friday, September 01, 2006

SetupBuilder for Clarion Anniversary Special

Quite a few users have asked for a SetupBuilder for Clarion anniversary upgrade special. If you are a registered user of SetupBuilder for Clarion" you can now switch to SetupBuilder 5.5 Developer Edition and get a full year of free upgrades and updates for $199.00 (save $100.00). Please enter the following coupon number when ordering online (Personal Information page): SWITCHNOW You will see your "Coupon (SWITCHNOW)" discount (-$100.00) reflected in the order summary at the right of the Billing Information page. Please do not continue if you do not see this SWITCHNOW coupon discount.
Posted Friday, September 01, 2006

Organize365 Price Increase

Organize365 is a Personal Information Manager and includes an API that Clarion developers can plug into free of charge. On Monday Sept 4 the price will increase from the current $29 per user up to $36.50. Organize365 includes a built-in peer-to-peer replication engine which allows users to share calendars and address books over a LAN automatically, without being dependent on a server.
Posted Friday, September 01, 2006

SetupBuilder Anniversary Sale

It's one year since SetupBuilder's first public Version 5 ("The Next Installer Generation") release. To celebrate the anniversary, LinderSoft has a special offer for Clarion developers. Buy SetupBuilder 5.5 Developer Edition for $299.00 and get six months of free upgrades and updates. This offer expires on September 5th, 2006.
Posted Thursday, August 31, 2006

Free Utility: BackItUp

BackItUp lets you quickly execute a backup and creates a nice .ZIP file at the location you specify. What makes it special is that within your backup profile you do not choose folders or files to backup, but instead you specify a starting path (and optionaly tell it to search the subfolders) and have it look for specific file masks in that folder(s). These file masks determine what files will be written to the backup file. For instance if you specify *.exe as a file mask then all files with the extension EXE will be backed up. You can also use more complex masks, and you can add as many file masks to a profile as you like. This allows you to quickly backup specific files within a folder tree. BackItUp runs in the system tray and lets you quickly execute your backup.
Posted Thursday, August 31, 2006

UK Jobs Site Updated

Richard Rose has updated the UK jobs site; the main difference with this version is its now completely free to use.
Posted Thursday, August 31, 2006

Clarion and the War On Terror (Update)

Ozzie Paez does data analysis in support of the war on terror, with Clarion as one of his tools of choice. These articles describe the type of end product the Clarion facilitates.

Posted Thursday, August 31, 2006

Alexander Ivanovsky Back Sept 10

From 26th August to 10th September Alexander Ivanovsky will be unavailable. Please direct all questions to Evgeny Stefanenko.
Posted Thursday, August 31, 2006

Free ClaColor Utility

ClaColor is a small utility program by Lansrad that converts HTML colors (like you would get from a ColorPicker program such as ColorImpact) into colors that you can use in Clarion. Type or paste in a HEX color code and get a LONG in Clarion's color format. This can be a very useful tool when you are working with setting matching colors for 3rd party tools such as XPTaskpanel or simply matching some external color scheme that you like.
Posted Thursday, August 31, 2006

New Prices At ClarionWebHosting

ClarionWebHosting has updated its prices and has introduced some new MS SQL 2005 Express packages. You can now host your MS SQL databases online in your own instance of MS SQL Express and access it direct from a Clarion app.
Posted Thursday, August 31, 2006

Template Writing Course

Course 5 of the SoftVelocity Clarion Training On Demand series takes a new approach to teaching template writing. Sessions include: Template architecture and language; Essential template writing; Extension templates; Control templates; Procedure and code templates; Utility templates; Creating a full template set; Template management with the ABC classes.
Posted Tuesday, August 29, 2006

Professional Icons $29.95 In August

All the icon sets at icons-icons.com are on sale for $29.95, through the end of August. That's a savings of $20 on each set. The Roma set includes 396 unique images, shadowed & plain, in 5 sizes, 4 graphic formats - that adds up to 19,800 icons. $29.95 through Aug.

31. Glasso Icons is the new Vista-looking icon set, this set comes with 60+ icons now, and another hundred will be added soon (free upgrade). Blanks for icons and buttons are also included. The colors in this set have been exactly matched to the new Vista operating system colors, and the images have been optimized by hand so that they will look good in the small sizes. Only $29.95 thru Aug. 31. Ace Brights and Ace Lights sets are experiencing a surge in popularity. 317 unique images, each in 5 colors, 4 sizes, 4 graphic formats, and shadowed and plain: that's 63,400 icons for only $29.95 (save $20). Use the Brights set for a bold look, or the lights set for a more subdued feel. Can be mixed. And if you want some great backgrounds for your software or websites, you can pick up the professional backgrounds set for only $9.95 - save $10.
Posted Wednesday, August 23, 2006

Ace/Roma Icons Customers Upgrade Notice

To all customer of Ace Icons, who purchased an icon set before mid-April* of this year: If you have not received an upgrade notice, you might need to send Sue your current email address. There are free upgrades for customers who purchased any Ace or Roma icon sets in 2006, from the beginning of the year through mid-April. And there are reduced price upgrades ($19.95) for customers who purchased prior to 2006. The Roma Icon Set has 133 new images (6650 new icons) added to it. There are new icons for databases, documents, folders, printers, users and more. The Ace Brights & Ace Lights Icon sets have 795 new icons added to them. There are new icons for RTF, queries, trees, mail, shipping, construction, medical and more.
Posted Wednesday, August 23, 2006

Free PDF-Xchange PRO for Clarion Developers

One non-transferable end-user license to PDF-Xchange PRO is now available to any Clarion developer who owns a Licensed copy of Clarion, any version. This is a $69.50 value. Download the evaluation version and email John Verbeeten and confirm in writing who you are, your contact info and that you are a Clarion dev. If you like the product and wish to resell to your own clients, reseller discounts are available starting at 30%.
Posted Wednesday, August 23, 2006

GTL6.23

GTL6.23 is now available for download. In ensuring that targets could only be EXE, DLL or LIB, the developer (we won't point fingers, but you know who) also managed to disallow paths that contained a period. This bug is now squashed.
Posted Wednesday, August 23, 2006

iQ-XML 1.20

iQ-XML 1.20 is now available. It has a few fixes with regards to making the writer a little more faster, as well as new function. XML:SetFieldPicture instructs the writer to format all data being written out, either manually or by "XML:AddFromQueue" or "XML:AddFromGroup". The picture must be a valid Clarion Picture Token such as @d010-. This way, your queue can rename as LONGs (holding Clarion dates maybe for sorting) but still write it out properly in XML:
Posted Wednesday, August 23, 2006

PlugIT Explained

Gary James walks demonstrates how to find and fix a memory leak using PlugIT.
Posted Wednesday, August 23, 2006

DevDawn: Drop-Down Listbox Vs Select Window

Stu Andrews asks: Which do you prefer? Drop-Down Combo/Listbox or Select Button/Window? Is it situational, or do you consistently use one or the other?
Posted Wednesday, August 23, 2006

MS Fixes Bug Affecting EmailReport and EmailData

Microsoft has fixed the bug in Aprils "cumulative" OE patch, which trashed the Send button in EmailReport and EmailData. Unfortunately, this fix does not sneak into WinXP automatically so you should download and install it manually. You can either get the patch from Microsoft or you can download OEHotfix.exe from Adobe. Log in as GUEST. OEHotfix will install the MS patch and do all Registry changes.
Posted Thursday, August 17, 2006

J-Fax 1.21

J-Fax 1.21 now supports cover pages. You can design a cover page with your company logo etc on it using Microsoft's Coverpage Editor (included with MS Fax), and save that as a .cov file. You then tell J-Fax which .cov file you want to use, what your cover page title and text should be.
Posted Thursday, August 17, 2006

GTalk Review

GTalk has been upgraded; review at DevDawn.
Posted Thursday, August 17, 2006

Firebird 2.0 Release Candidate 4

Kelvin Chua reports that Firebird 2.0 Release Candidate 4 has been released.
Posted Thursday, August 17, 2006

PDF-XChange & Tools SDK New Release

An important update has now been released (Build 3.60104) which includes: PDF Forms Creation - see demo app pdform36.app; New and improved support for IceTips; A number of minor fixes and issues resolved. Please also note that a number of the revised/new templates and class files now sport the suffix '...36' to denote the new version update - as does the Clarion PDF manual for PDF-Tools : PDFToolsCW36.pdf.
Posted Thursday, August 17, 2006

WindowID 1.03

New in WindowID 1.03: Workaround for Clarion bug PTS 11315; You can now select the location of the WindowID button (combining top/bottom and left/right options); Global option to show current path information (optional in addition to current options); Global option to show procedure's parameter (optional in addition to current options); If the procedure description has quotes or special characters, the compiler issued an error, now it is supported; Procedure description no longer generate local code description if is not globally enabled; Global options to choose displayed information are no longer related to "Always active" prompt status. Note: if you upgrade from a previous version, a full recompilation is needed.
Posted Thursday, August 17, 2006

Dr. Explain At DevDawn

Over at Dev Dawn you can find a review of Dr. Explain.
Posted Thursday, August 17, 2006

Whitemarsh August Announcement

There are two main items for this month's announcement: The Data Interoperability

Community of Interest Handbook, and The Short Paper Series edition. The Data Interoperability Community of Interest Handbook is almost complete. It has been sent out to key persons in the industry who work in data interoperability intensive communities of interest. These experts' suggestions will be folded into the draft and then a finalized first edition will be published in September. The second item is the new short paper: Managing Data Models . This paper is an extension of the three previous publications, concentrating on the management of data models. The Managing Data Models paper posits that managing data models through transformations is not very productive. Rather, data models should be managed through where-used mappings. Keep in mind, the complete short paper series publications are available.
Posted Thursday, August 17, 2006

Gitano Software Affiliate Program

There's no charge to join the Gitano Software Affiliate Program. Place a banner, text or other designated link on your site or home page and Gitano will pay you a commission on all sales generated through your site. Receive a 15% commission (or more) on each purchase your visitor makes after clicking on the banner or other custom affiliate link. The program is available for Gitano Software Utilities, 1st Icon Design Icon Sets, 1st Logo Design Stock items like Logos, Web Templates etc., Custom graphic design from 1st icon Design and 1st Logo Design (this option will require specific listing in your site for this service). Click on the Affiliate log in menu option.
Posted Thursday, August 17, 2006

SetupBuilder 5.5 Developer Edition

SetupBuilder 5.5 Developer Edition has been released. SetupBuilder(TM) is a powerful and easy to use Installation Authoring and Configuration Management environment for developers and organizations deploying applications to the Microsoft Windows Platform, including Windows x64, Windows 2003 and Windows Vista. SetupBuilder 5.5 Developer Edition starts at $299 USD for a royalty-free usage license. A trial version is available.
Posted Thursday, August 17, 2006

Huenuleufu Biggest Dictionary Contest

Huenuleufu is having a contest to find the biggest Clarion dictionary. Pick your biggest DCT, export it to TXD, ZIP it, and email it in. The biggest dictionary earns a choice of US$50 in rebates, or one of Huenuleufu's products free (up to a US$99 value), or a two year maintenance plan (up to a US$120 value). Dictionaries will be used for testing to improve the FullRecord product. The contest ends August 30th (2006).

Posted Thursday, August 17, 2006

Clarion Magazine

OSCON: Query Interfaces, PHP/Web 2.0, and TimeTravel Tables

by Dave Harms

Published 2006-07-29

Building "Sane" Query Interfaces to Databases - Mark Newsome

Mark Newsome, an HP developer, presented his take on improving query interfaces for web applications. His interface starts with a search form; the user chooses zero to n filter options, and then clicks a button. Rather than immediately displaying the search results, which could be in the thousands of records, he displays the number of records found, allowing the user to further refine the query if desired.

Keys to the search interface's ease of use are:

- Dynamically creating popups for shorter lists of filter lookup values
- Converting popups to lookup buttons where filter item lists are long
- Pre-qualifying date ranges by showing the min and max in the available data
- Validating on the client side where practical, to reduce trips to the server
- Converting * to % for SQL (users are more familiar with the * wildcard)
- Trimming string data on entry into database

Mark also commented on the user of cursors for large data sets. The problem here is that you have to keep the cursor open, but since this is a web app you have no idea how long the user will be in the session. You might be using that resource unnecessarily. Instead, you can keep a list of just the primary keys (and maybe a sort field) for the returned items, in the sort order, and when needed for another page retrieve the rest of the data by primary key.

Rasmus Lerdorf - PHP and Web 2.0

Rasmus Lerdorf, the inventor of PHP (and now one of about a thousand PHP committers), described Web 2.0 as putting a prettier face on the web, with fewer visible round trips to and from the server (there may actually be *more* trips but the user doesn't see most of these). You do need a rich presentation layer language, a server-side language (PHP in Rasmus' example, but it could be any of a number of other languages), a wire data format (XML and or JSON), a server side data store, and a semblance of security (there is *no such thing* as a totally secure web app).

Rasmus presented a scenario where a web site has half a million users, at an average of 100 requests per use. This works out to 578 requests per second, and you should use a factor of three to determine peak load (1734 requests per second).

You can load test with http_load from acme.com. If your CPU idle time is zero and you're not at your target, you can work on speeding up your app. If your CPU is not bound, then you are IO bound.

20 ms is a good target for average latency - 250 ms is pretty bad. The initial version of the app could only server 17 requests per second, meaning the application would need 100 servers, a substantial expense for a company in startup mode.

Rasmus commented that "if you wrote it, you're the last person who should be fixing it" since you would have written it better if you knew how. You can analyze your PHP using the Callgrind profiler, which sits on top of Valgrind. Run your server under Valgrind/Callgrind for testing only as profiling slows the server tremendously.

Rasmus demonstrated how non-persistent, secure connections to PostgreSQLcan significantly hurt performance, and noted that MySQL is dramatically fast at connecting; non-persistent MySQL connections are faster than persistent connections in PostgreSQL. Query caching boosts performance a little more.

After pooling/caching database connections, the next step is to add an opcode cache like APC, so the server doesn't keep reloading your PHP code.

Eventually Rasmus got the app trimmed to a very respectable 1100 requests per second, for a two-server system instead of a 100 server farm.

Rasmus also demonstrated some useful features in PHP 5, including including, reluctantly, SOAP capability. He's no fan of SOAP, demonstrating the massive WSDL file needed for a service that simply adds two numbers. He is a fan of RSS. Other nice bits in the presentation included geocoding and map tiling APIs from Yahoo. He concluded with a 30 second AJAX demonstration using Yahoo controls, and pointed to the Yahoo User Interface Library page.

You can download the slide show online. The slides were available online at the start of the session, and with wireless working it was nice to be able to view my own copy during the presentation.

TimeTravel Tables in PostgreSQL - A. Elein Mustain

TimeTravel tables allow you to get snapshots of your data at a particular point in time. Although A. Elein demonstrated this technique for PostgreSQL, you can apply the concept to just about any SQL (or, with extra code, a flat file) database.

The basic cost to your data structure is two timestamp columns in every row; queries require a qualification on one or both columns, and there is a small overhead for update and delete triggers.

In the old days PostreSQL did have time travel, since it had no-overwrite storage and you could simply decide not to vacuum the database. This technique creates the same kind of storage. You only insert data - deletes don't remove data, and updates insert a new changed row. You can be selective about which tables use this technique; in PostgreSQL you can template the necessary code.

First, define a table with a primary key, with start and end time columns (timestamps). The combination of primary key and end time must be unique. Current data will always have a null end time.

Create an `inventory_at_time(timestamp)` function for each table (in PostgreSQL, one copy of the function for each table results in faster, inline SQL) which returns a set of records which are valid for the given timestamp. You can further use this set in a FROM clause.

The Delete rule simply sets the end time to the current time stamp where the key matches and the end time is null.

Updates are more complicated. If someone else is updating the row, the update needs to be disallowed. If a non-null end time is present on the old record then do nothing. If the new end time is null, then update the old record end time with the current timestamp, and insert the new record.

You may or may not want to be aggressive with the insert trigger, enforcing null end time and current start time.

Using this technique, you can retrieve a snapshot of your data at any time.

A. Elein Mustain is the author of an irregularly weekly column called PostgreSQL General Bits.

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

Reader Comments

Add a comment

Clarion Magazine

OSCON: solidDB Beefs Up MySQL OLTP Choices

by Dave Harms

Published 2006-07-28

The open source database landscape has shifted somewhat in the last year, with Oracle on a buying rampage, picking up SleepyCat (Berkeley DB) and InnoDB, both of which provide back end file systems for MySQL. Berkeley DB is a flat file system, and InnoDB adds transaction processing and other capabilities to MySQL. Oracle also tried to buy MySQL AB but was rebuffed.

The sale of InnoDB to Oracle has left some MySQL users with concerns over the long-term availability of transaction support, despite Oracle's assurances that it will continue Innobase's relationship with MySQL.

Seeking an opportunity to enter the open source marketplace, high end database provider Solid Information Technology has stepped up with a version of its OLTP database for MySQL, released in beta on July 25.

Solid Information Technology has a track record in the telecommunications industry. Founded in 1992 in Finland, the company has sold mission critical databases predominantly to network equipment manufacturers like Nortel and Siemens. The first product shipped in 1994, and as Senior Product Manager Murat Demiroglu notes, all the original developers are still with the company. There are more than three million Solid database installations worldwide.

Features of the proprietary version include sub-second failover, high scalability, and high reliability. An SQL product since 1994, Solid's OLTP database has the expected mainstream features such as triggers, stored procedures, and events. ODBC and JDBC drivers are also available for the proprietary product.

Solid's version of its database engine for MySQL is licensed under the GPL, and uses MySQL's pluggable architecture. Access is through MySQL's front end, using drivers/providers already available for MySQL.

In an interview at OSCON 2006, Demiroglu made the point that high availability web sites have similar OLTP needs to those in the telephone industry, such as many short transactions and a high level of concurrency. solidDB offers optimistic and pessimistic concurrency, and row level locking. A unique indexing technique called the Bonsai Tree makes multiversion concurrency control possible for reads as well as rights which improves scalability. Partitioning of databases is scheduled for MySQL 5.1, at which time solidDB will take advantage of this capability.

In a low end environment, with one CPU, InnoDB is admittedly faster. But Solid's target market is larger, mission critical systems with high concurrency, many transactions, and multiple processors. solidDB is designed to excel in these environments.

The solidDB storage engine is currently bundled with MySQL 5.0, and source code is now included. Releases are available for Windows and Linux, and more platforms will be added over time.

The licensing arrangement between MySQL and Solid was helped along by close proximity - both companies are located in the same Cupertino office building.

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

Clarion Magazine

OSCON: Wednesday Keynotes

Published 2006-07-27

Wednesday's keynote address was really a series of short keynotes, introduced by O'Reilly's Nathan Torkington, an informally dressed and chatty Kiwi with a fond memory of past OSCON blasts. Nat shared stories of luminaries at past OSCONs, including two thirds of the family Dyson, including son George. I'm particularly sorry I missed that conference, as some years ago I lived across the water from the Belcarra treehouse that was once George Dyson's home, as described in The Starship and the Canoe.

Conference organizer Tim O'Reilly (well, Nat and his team actually do the work) kicked things off with a few points on the open source phenomenon. He discussed Web 2.0's architecture of participation, and pointed out that in a Web 2.0 world, open source licenses are obsolete since end users typically don't need licenses to use web apps. He also mentioned the phenomenon of asymmetric competition, noting that the #7 site on the net, Craiglist.org, has 18 employees, while the rest have thousands (Disney, at #8, has 129,000).

Some open source frameworks, O'Reilly pointed out, have come out of closed source projects. Both Django and Ruby started as proprietary applications, and the frameworks were extracted.

O’Reilly is a fan of FireFox plugins, Asterisk (the open source PsBX), and the Ubunto Linux distro. He also plugged labs.oreilly.com, where you can search some 2.6 million lines of code from O’Reilly books.

O'Reilly was followed by Greenplum's Scott Yara, who presented an amusing comparison between open source and rock'n roll, asking if they were twins separated at birth, and admonishing attendees not to get stuck in "we're huge in Belgium" mode. Anil Dash of Six Apart (TypePad, MovableType, VOX, LiveJournal) ripped through an entertaining talk titled "Trying to Suck Less: Making Web 2.0 Mean Something."

The final keynote as by Mike Olson of SleepyCat, another recent Oracle acquisition. When asked if PostgreSQL or MySQL were looming acquisition targets, he replied that PostgreSQL's licensem ade acquisition almost impossible, and he alluded to Oracle's previous unsuccessful attempt to buy MySQL.

The keynote session concluded with a panel Q&A. From left: Nat Torkington, Tim O'Reilly, Anil Dash, Scott Yara, Mike Olson.

Clarion Magazine

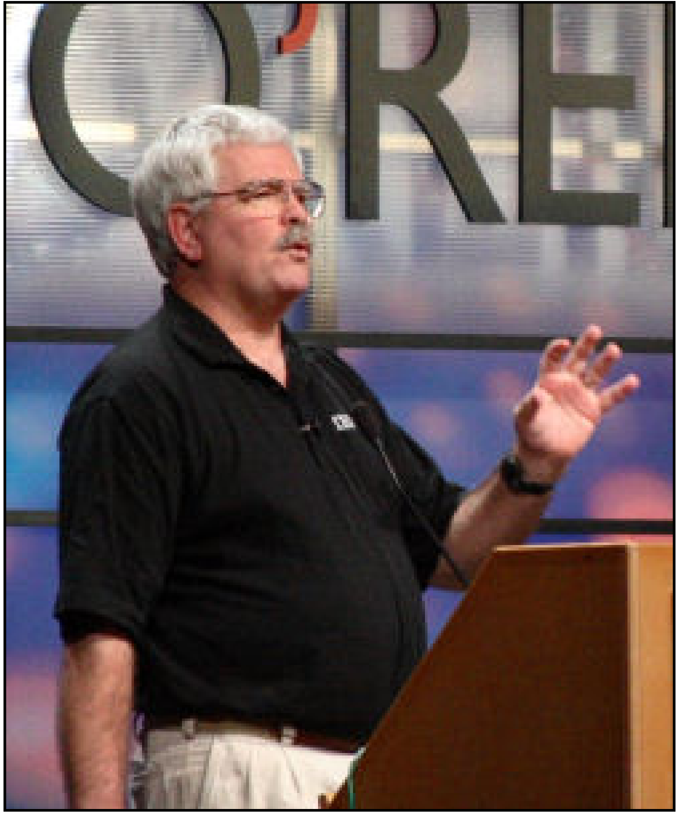
OSCON: PC History, Tech Trends, and Wrap-up

by Dave Harms

Published 2006-08-09

25 years of the IBM PC

Dave Bradley is the author of the original IBM PC BIOS, and the man who invented the three fingered salute. He gave a few highlights of his session later that morning, in which he covered the history of not just the IBM PC, but the personal computer industry in the years leading up to the introduction of the IBM PC. Key figures included John Draper, AKA Captain Crunch, who discovered that the whistle in a box of Captain Crunch would trigger access to the telephone long distance switching system. Draper was also the author of EasyWriter, which was written for IBM.



Bradley showed a picture of Microsoft's Donkey, the last program Bill Gates wrote that got shipped with any product, and which Bradley thinks of as Bill's lasting legacy.

In testimony to the relative youth of the audience, there was absolutely no crowd reaction to the photo of the PC Jr.'s "chiclet" keyboard.

Bradley commented several times on the inability of developers to foresee the future. Often a short meeting or a few minutes work resulted in something that lasted for decades. The original character set of the IBM PC was the result of one five hour meeting among three people on an airplane. Ctrl-Alt-Del took six lines of code, and five minutes of work, and was never intended for the general public. The same goes for marketing - in 1981, IBM sales projections over the five year lifetime of the IBM PC were for 241,683 sales. Actual sales ended up at around three million.

Bradley concluded with the admonition "Whatever you do, don't forget to have fun."

Technology Trendspotting with the O'Reilly Research Data Mart

Roger Magoulas presented a stack of stastics from O'Reilly's research department, including the Bookscan treemap. It should come as no surprise that technology is changing the book industry; what may be surprising is that tech book sales have been flat since 2004. I found it difficult to draw conclusions from what sometimes seemed to be conflicting data, but a few points stood out:

- Java and .NET seem to be similarly sized markets, with about 15% overlap
- JavaScript (presumably due to AJAX) and Ruby on Rails are on the way up
- JavaScript has surpassed PHP in book sales
- Web 2.0 a big factor in job postings
- There is a red state/blue state thing happening with open source/closed source
- SQL server book sales are up
- MySQL and PostgreSQL aren't showing much growth, but MySQ L is still much bigger than PostgreSQL
- VOIP/Asterisk is a trend to watch
- Virtualization also something to watch
- iPod/iTunes book sales are up

The Washington Bridge and Other Public Structures (Hacking Government)

Carl Malamud is an open source advocate and political animal who gave a sometimes disjointed and occasionally very funny presentation on lobbying all levels of government in the interest of open source. Unfortunately I just about fell asleep during the presentation, due to a combination of Friday morning fatigue and my relative unfamiliarity with the US political landscape.

openTalk 2.0: Maximizing Non-stakeholder Buy-in by Leveraging Depatented Generic Information Transfer Protocols

I gather Damian Conway has a real job, but if it isn't standup comedian, it should be. Conway, introduced as a paragon of the open source community who had inexplicably signed on to a mysterious startup, bounded on stage in suit and sunglasses to announce openTalk 2.0. Confirming the theme of the conference ("we have won!"), he announced the open source has "jumped the shark and banged the monkey." He then proceeded to announce a series of upcoming products, their names complete with key letters replaced with other arbitrary letters. These include cxap (pronounced "crap") and a micropatent micropayment system called paxo (pronounced passgo)."Every time you paxo, we collect $200." Other products included cxap mapx, for maps just like you'd draw yourself, and cxapr - put all your patents in the cxapr and cxap will administer them for you. cxap also has patent pending on the technique of invalidating patents by demonstrating prior art.

And what would the cxap suite be without cxap on rails, which you can use to turn any existing web site into the web site you want. Conway combined "pr0n" with the Google site, and a short time and numerous credit card payments later had a new web site - go.ogle.com. And of course cxap on rails uses HECTOR, which kicks AJAX's butt.

I really hope this session finds its way to YouTube.

Keynote panel, from left:
Nat Torkington, Carl Malamud, Damian Conway, Roger Magoulas, Dave Bradley

Closing Keynote: Eben Moglen

Eben Moglen is General Counsel of the Free Software Foundation, and an eloquent advocate of open source licenses. He is also currently at the center of a controversy over digital rights management and GPLv3, the next version of the GNU/General Public License. Linus Torvalds has been critical of some of the changes in GPLv3, and has said that Linux will continue to be released under GPLv2.

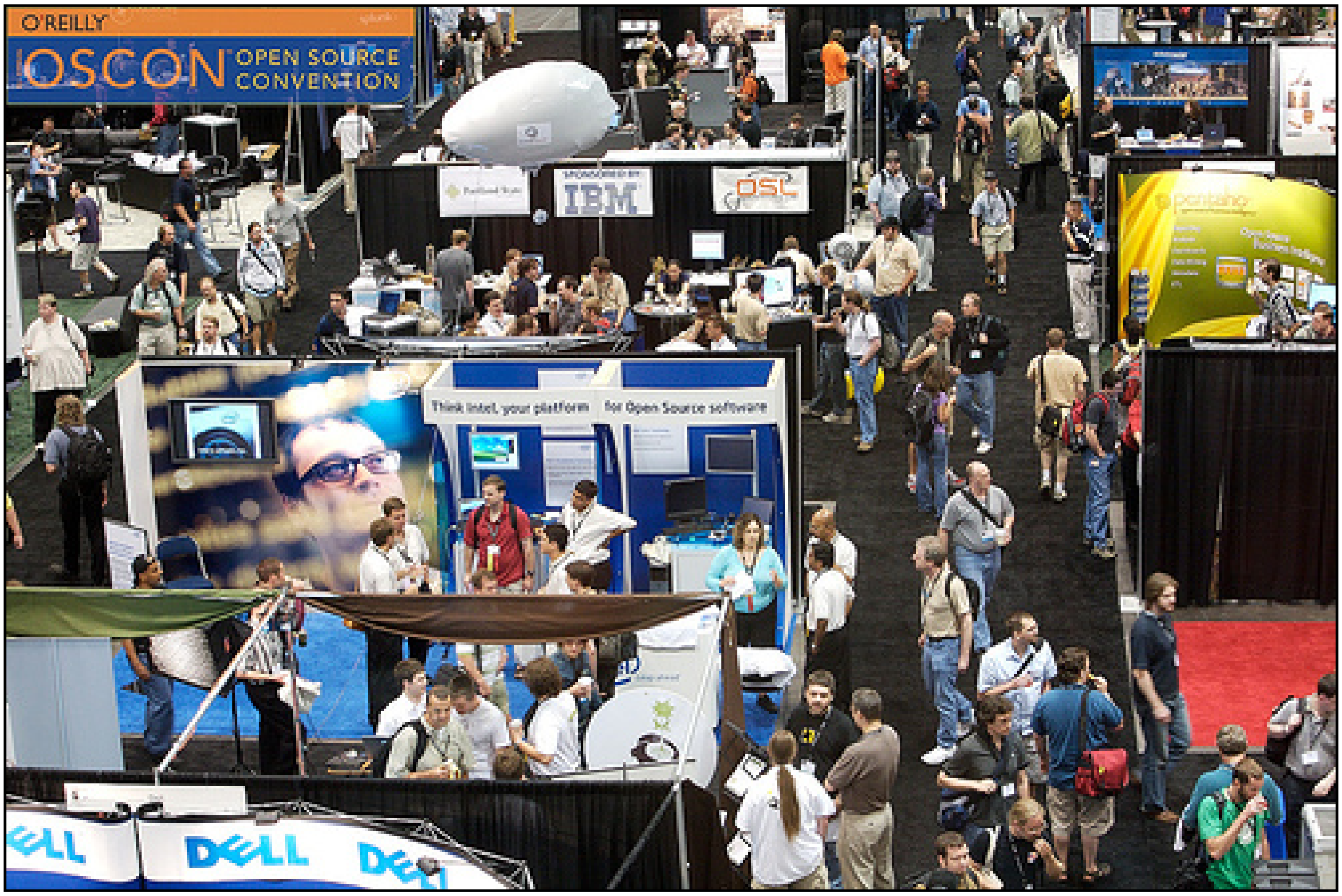
For the details on Moglen's closing address I'll refer you to Joe Bronkmeier's NewsForge piece. You'll notice the mention of the standing ovation. Although I'm clearly a fan of open source, I don't think it's the answer to all the world's ills. I like r0ml's target of more than one third, less than two thirds open source. As a relative outsider, that standing O, coming after the conference's oft-struck note of "We have won!" struck me as the week's "drinking the Kool-Aid" moment. Not that there's anything wrong with that.

OSCON for Clarion Developers

If I'm not back at OSCON next year, it will most likely be because of an already-looming scheduling conflict. I'd be happy to take in this conference again. The keynotes were informative and entertaining (sometimes both!), the track sessions numerous and varied. I focused on the database track, but there were a lot of other sessions I'm sorry I missed, particularly those on open source web development tools. Obviously there were no Clarion-specific sessions (in addition to being an open source conference, this is largely a Linux show) but the principles of database-oriented development remain constant across deployment platforms, and I came away with some new ideas and approaches.

If you're interested in broadening your database horizons, and/or you're working with open source tools, I suggest you consider attending next year's OSCON. If you do any work with the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python) or Ruby, OSCON is definitely going to be worth your while.

The next OSCON is again in Portland at the Oregon Convention Center, July 23-27, 2007.



OSCON Exhibit Hall
Photo credit: James Duncan Davidson/O'Reilly Media

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

Reader Comments

Add a comment

Clarion Magazine

OSCON: Keynotes, Database Tuning, and .NET Tools

by Dave Harms

Published 2006-08-08

Robert Lefkowitz

Robert "r0ml" Lefkowitz is the Chief Technical Architect at AT&T Wireless, and an advocate of open source initiatives in large enterprises. When preparing his presentation, r0ml was faced with the need to complete the sentence "Open source is like …" And what came to mind was… "a tomato." As a man famous for his metaphors, any number of possibilities came to mind. He listed one after the other, and each time adamantly declared “but I’m not going to make *that* metaphor.” The metaphor he eventually settled on was that the tomato, when adjusted for consumption rates, is the most important source of vitamins A & C and an important source of vitamins and minerals.

How much OS is healthy? “Some people say it should all be open source, and these of course are the vegetarians, or vegans, I suppose.”

As a “reasonable, rational, American businessman” r0ml would like to see at least one third of software open source, and probably no more than two thirds.

r0ml blogs at Taking IT Personally.

Stephen O'Grady

Stephen O'Grady made the point (not for the first or last time at this conference) that "you guys have won." And with that power comes responsibility. Key points included loweng barriers to entry (better docs), being creative, and working together.

Jason Scott

Jason Scott showed a trailer for his upcoming documentary, Get Lamp, the story of text adventure games (shot, naturally, in HDTV). Scott made the point that the point that a lot of computer pioneers have been been forgotten by mainstream historians. For instance, he conducted the first ever video interview with Ward Christenson, the founder of the first ever BBS, and the author of the XMODEM protocol. Scott encouraged attendees to document, even in small ways, the work they are doing now.

Google's Open Source Project Hosting Service

Probably the most-anticipated session of the conference was Google's "TBA" presentation, rumored to be the company's announcement of an open source hosting service similar to that offered by SourceForge. The meeting venue was standing room only; at one point, before the session started, someone called out to Google's Greg Stein, "You can start now!" To which another attendee replied "He's not done coding yet."

Greg Stein, an engineering manager at Google and chair of the Apache Software Foundation, began his presentation with a list of recent Google programs, including the Summer of Code, funding and grants, partnering with FireFox, Google Earth, and the release of various bits of code. So what's next?

As expected, Stein introduced not a dating service for geeks, but an AJAX-based project hosting service for open source projects

In support of this project Google has rebuilt Subversion on BigTable, which is a "massively scalable failover blah blah blah (technical term - ed.) storage system." The Google team also did a rethink of issue tracking. There is a lot of workflow structure in current issue tracking systems. Much of this has been replaced in Google's product with labels, which can be tracked using the Google search code. Ironically neither BigTable nor the Google search code are open source, however.

There is no approval process for projects; project developers only need a Gmail account. But there are monitoring algorithms that will detect unusual usage patterns, such as massive uploads/downloads not normally associated with open source projects.

The following demo focused on the issue tracker aspect. As the project manager you decide the basic fields and labels that will be used in the issue tracking system. These are key/value pairs, but they don't restrict the user. The use of drop-down combos makes it possible for the user to override the default values when necessary.

At a minimum issue data will be available as XML, but there should be an API eventually. These kinds of APIs will be delayed as long as possible because once built they tend to get locked in.

Automatically emailing issue items is not yet functional, but "every application expands to the point where it can send email."

Google groups are used for discussion - and Google groups are email lists. You can also add blogs and other links to your project, as well as existing mailing lists.

There are only seven licenses from which you can choose, specifically to avoid license proliferation. These include Apache 2.0, Artistic/GPLv2, Mozilla, MIT, BSD (without advetising), GPL2, and LGPL. GPL3 will be added. Developers are strongly encouraged to stick with a single license, but it is possible to change a project's license at any time.

Stein said Google wasn't taking aim at the other open source project hosting sites such as SourceForge. The names of all 166,000 SourceForge projects have already been reserved at Google to prevent accidental and malicious name duplication. Google would like to do the same with the other sites. As well, projects on other sites can, if they wish, use only part of the Google hosting system, such the issue tracker.

Attendees were reminded that the system is in early beta. There is right now no way to download tarballs, although you can get around this by checking in a tarbarll.

One of the folks from opensource.org announced a license choosing wizard. I wasn't at that session and couldn't find a link on their site, but OSSF does have a wizard online.

Eight Steps to Fix Your Database Performance Problem - Christopher Browne

Christopher Browne's session focused on PostgreSQL performance issues on Linux, but many of the techniques mentioned can be applied to other databases on other platforms.

Logs can offer valuable information; see `/var/log/messages` for hints about failing hadware, and the database logs for database activity. Application logs can provide yet another perspective, and their contents may not always agree with the database logs (as when the app log reports a data operation takes longer than the database log says because the app has some extra overhead).

You can set PostreSQL's `log_min_duration_statement = 1000` so any query of over a second will show up in the log. You can also turn on logging for all queries by setting = 0 but that could swamp your system.

PgFouine is a Postgres log analyzer - gives you summaries that let you find slow and frequent (aggregate time) queries.

Resource utilization falls into three categories:

1. RAM
2. Disk I/O - updates force writes, poor caching forces reads (as the cache itself spills over onto disk)
3. Table locks - a good way to bring an app to a halt.

CPU bottlenecks are relatively rare, and when it happens is often a symptom of I/O problems. You could be swapping or writing to disk excessively.

Network misconfiguration causes all kinds of evil. Tools to use include: top, vmstat, iostat, sar.

Database configuration defaults are seldom optimal. Postgres used to be particularly bad for this, but when changing settings always use a light hand - don't crank everything up to the max.

Useful PostgreSQL settings include:

- `shared_buffers` - Controls use of shared memory. Shouldn't exceed 15% of memory or 10K
- `effective_cache_size` - How much OS cache do you have?
- `work_mem` - How much memory to use for each sort/aggregate? If you set this too high you'll get into swapping problems.
- `maintentance_work_mem` - How much to use with vacuum?
- `max_fsm_pages`/`max_fsm_relations` - Cheap to use, how much mem to use to allow deferring vacuum
- `fsnc` - Turn off for high speed/high risk situations. Okay when filling a database, but not for regular use.
- `checkpoint_segments` - This is the distance between write-ahead segments. Increase it if you see a lot of updates in the logs
- `random_page_cost = 4` - How much more expensive are reandom scans than seq scans? Set to less than 4 if you have good disks.

- `temp_buffers` - Iincrease this if you use a lot of temporary tables

Make sure the server doesn't read a lot of empty space. In MySQL use `OPTIMIZE TABLE`, in PostgreSQL use `VACUUM FULL`. Typically this kind of operation locks tables.

Do some table analysis. You can use a statement like `ALTER TABLE x ALTER COLUMN y SET STATISTICS 100` to set the numbe of bins used for analysis. Use `EXPLAIN` / `EXPLAIN ANALYZE` to discover what the server is actually doing with your queries.

Keep in mind that if you have one disk at 7200 RPM you can't do more tan 120 ransactions per second, because you only get one write per disk rotation. RAID increases this, and you can add a $2K RAID controller with battery backed cache for a "cheap turbocharger." You can use tablespaces to separate data into multiple partitions, but this adds the complexity of management by humans.

Reuse your connections because they're expensive. Application frameworks frequently offer connection pools. `pgpool` is a connection pool proxy.

.NET tools

In one of the few Windows-oriented open source session, Lee Fisher tore through a seemingly endless list of .NET-related tools at breakneck speed, pacing restlessly between computer and projection screen. At one point he paused to take a breath and asked the audience for the time. Told he had half an hour left, he responded "Oh geez, I gotta go faster."

You can download Lee's list here. Based on Lee's commentary, I'd say the higher up a product is in each list, the better it's likely to be, but you can't take that as a hard and fast rule.

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

Reader Comments

Add a comment

Clarion Magazine

ClarionMag at OSCON 2006

by Dave Harms

Published 2006-07-26

This week Clarion Magazine is at OSCON, O'Reilly's Open Source Convention held annually in Portland, Oregon.

You may be wondering what ClarionMag is doing at an open source convention, given that Clarion is a closed source development tool. First and foremost, I'm here to report on the current state of open source databases of interest to Clarion developers, including MySQL and PostgreSQL. And as time permits, I'll also be reporting on some open source web development tools. Although Clarion developers do have some Clarion code options in this area, tools like PHP and Ruby are becoming increasingly popular. OSCON was a Perl conference until 1999 when its scope was widened, and Perl is still a big part of the culture here.

As usual, I experienced a few minor disturbances of my personal space-time continuum on my way to the conference. We did a little holidaying beforehand, and in the rush to get everything packed I neglected to include any business cards. Well, no worries, I was overdue for some new cards anyway, having long-since changed the look of the ClarionMag web site, complete with a new logo by Frostbytes.

The only problem was I realized all this on Saturday night in Langley, BC (1500 miles from home), and the conference sessions began bright and early Wednesday morning, another 300 miles away. I had two working days to get the cards in hand. My graphic designer, Leroy Shulz, had the gall to go rock climbing over the weekend, so on Sunday morning (our driving day to Portland) I cobbled together a basic card with the new logo. I also did some digging for a Portland printing company and found a recommendation for Phoenix Media. Their web site said they could print and ship cards in 18-24 hours. All I needed was local pickup, and the web site said that could be done around the clock. All seemed fine.

Monday morning, as I was getting ready to upload the card graphics, Leroy popped up on Skype, and an hour later I had a very fine card design in hand. I fired the file over to Phoenix Media, and followed up with a phone call. The job would be ready by 5 p.m. Tuesday, and I could pick up the cards any time before 9 p.m.

The only problem with that plan was that Tuesday's schedule called for an afternoon boat ride on the Willamette River to a floating restaurant in downtown Portland, and then back to the boat ramp somewhere around 9 p.m. No time to drive over and get the cards.

A quick check of the map revealed that the restaurant wasn't that far from the print shop, so between the end of the meal and the time the check arrived I hopped in a cab and made the pickup. Mission accomplished! After dinner we had a relaxing cruise back up the river, passing a whole lot of luxury homes, a fair number of architecturally unique floating homes, and not a few water skiers, wakeboarders, tubers and jet skiers. The Willamette south of Portland is a lively place. Portland remains one of my favorite cities, and it was a treat to see a side of it I never knew existed. Many thanks to Bruce and Wanda for the trip!

OSCON is being held at the Oregon Convention Center, which is across the river from downtown. This is fairly modern facility, with plenty of room for this crowd. Seating for the keynote addresses looked to be around 1500-2000, and the seats were about two thirds full for the Wednesday morning session.

As I mentioned earlier, I'm mainly following the database track, but there are many other tracks, including:

- Javascript/AJAX
- Java
- Business
- Web apps
- Ruby
- Security
- Python
- Programming
- PHP
- Perl
- Windows
- Linux
- Emerging topics
- Desktop apps

What I find interesting so far about the database sessions is the relative invisibility of Ingres and Firebird. The omission of Ingres I can understand, as that's an older technology only recently released as open source. But Firebird, which has enjoyed some popularity among Clarion developers, isn't getting any traction here, as near as I can tell. PostgreSQL and MySQL, however, both have several dedicated sessions, and PostgreSQL.org has a vendor booth as well. Developers from both camps are also in attendance. I'll have more to report as these sessions come up.

Demographically this is a more varied group than the largely middle-aged (and up) male crowd you’ll find at a Clarion conference, and that’s a nice change. There’s an idealistic vibe here that I haven’t seen among Clarion developers in recent years, and some of that may be due to the younger crowd. But even the old guys were busy during the Wednesday morning keynotes admonishing the crowd to “do what you love” and “make the world a better place.” And that means getting down to the code. Overheard in the halls:

> “Where are your peeps, man?”
>
> “They’re writing code like there’s no tomorrow.”
>
> “Why, is there no tomorrow?”

I sure hope there's a tomorrow. Lots of good stuff scheduled. And I'll have a report on today's sessions up soon.

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

Reader Comments

Add a comment

Clarion Magazine

Marking Time 3: Inter-Date Computations

by Steven Parker

Published 2006-07-20

Basic date math, such as adding days to a date, or finding the number of days between two dates, is easy in Clarion. Time math gets a little trickier, because of the way Clarion assigns time values. And things really start to get interesting when calculations involve midnight rollover and other operations involving date/time values on different dates.

Date arithmetic

Because Clarion stores dates as Longs, date arithmetic is simple arithmetic. For example:

```
ElapsedDays = EndDate - BeginDate
TotalDays = EndDate - BeginDate + 1
FutureDate = DateVariable + numberOfDays
```

My favorite is:

```
xDate = Deformat(20040419,@d12)
If (Today() - xDate) % 14 = 0
  RecurringProcedure
End
```

This code allows me to perform `RecurringProcedure` every 14 days, precisely on the anniversary of 19 April 2004.

Likewise, because Clarion stores time of day as a Long, time arithmetic is also very easy.

But, because midnight is defined as "1" (there is no zero in Clarion time), there is an inherent offset of one (see The Clarion Advisor - Calculating Times for a complete discussion of handling the issues this can and does cause):

```
ElapsedTime = (Clock() – 1) – (BeginTime – 1) + 1
```

And, because the result of a time computation is a numeric, computing and displaying elapsed time or displaying a countdown timer is *just* arithmetic. See Marking Time, Part 1, Marking Time: Round 2 and Another Approach to Time Calculations to see various approaches to computing and display time differences.

Time computation is a very popular subject in the Clarion world. I have cited four articles but there are more. And despite this fairly extensive coverage, questions about computing the difference between two dates and times recur on the newsgroups with almost predictable regularity.

Midnight rollover

"Midnight rollover" is the term that has come to describe a condition requiring comparison of times on different days. When doing time computations, it is necessary to understand how to determine whether one (or more) midnights, days, have passed. It is not safe to simply assume the two times are on the same day. Neither is it safe to assume that, if there has been a midnight rollover, only one day has passed. In fact, it isn’t even safe to assume that there has indeed been a midnight rollover.

To get a handle on the issues, let’s start by thinking in terms of four variables (all Longs):

```
BeginDate
BeginTime
EndDate
EndTime
```

Assume that `BeginDate` and `BeginTime` were assigned values programmatically. The ending date and time are often just `Today()` and `Clock()`. They could also have received values programmatically, for example, in a time clock application.

I want to be able to determine the difference between `EndTime` and `BeginTime`. So first, I need to determine whether or not one or more days have passed. This part is simple enough, just compare `BeginDate` and `EndDate`.

There are exactly three possibilities. Either `EndDate` is greater than `BeginDate`, the two are equal or `BeginDate` is greater than `EndDate`.

`If EndDate = BeginDate`, the dates are the same, there has been no midnight rollover and

```
    ElapsedTime = (EndTime – 1) – (BeginTime – 1) + 1
```

gives me the information I need. I can convert `ElapsedTime` to seconds (suppose I want to do something when my program has been idle for a certain number of seconds), minutes or hours (an employee time clock application, for example). I can convert `ElapsedTime` to show how much time has elapsed. Etc.

The remaining two conditions require programmer action.

`If BeginDate > EndDate`, this is an elapsed time condition that should never occur. There are only three things I can think off to cause this: (1) a user resetting her/his clock, (2) a failing system clock (motherboard failure) or (3) an "SPE" (stupid programmer error). After eliminating (3), any other time `If BeginDate > EndDate`, I would think in terms of a very noticeable message and summarily terminating the application.

That leaves `If EndDate > BeginDate`. Unfortunately, this isn't quite straightforward.

While `EndDate – BeginDate` tells me how many days have elapsed (and I can multiply to get the number of hours, minutes or seconds), the time variables cannot be directly compared any more. This fact is obvious when considering a time clock. Suppose an employee clocks in at 23:00 and clocks out at 07:00. `EndTime` will be greater than `BeginTime`. So instead of translating to eight hours, it will compute as a negative number (-576000 ticks), an impossibility. If I try to be clever and use the absolute value or try to subtract the smaller number from the larger, I get 16 hours.

Why? 23:00 (11:00 P.M.) is 8,280,000 clock ticks. 07:00 is 2,520,000 clock ticks. The difference is -5,760,000 which translates back to -16 hours. It should be eight, of course.

The situation would be even worse for an employee clocking in at noon on Tuesday and clocking out at 17:00 Wednesday. This would give seven hours instead of 31.

If I add one day (8,640,000) to `EndTime`, making the computation:

```
  11,160,000
 - 8,280,000
 ===========
   2,880,000
```

I get a correct translation to eight hours.

So, a completely general codelet would look like:

```
If EndDate = BeginDate
  ElapsedTime = (EndTime - 1) - (BeginTime - 1) + 1
Elsif EndDate < BeginDate
  Stop('Invalid dates!')
Else
  ElapsedDays = EndDate - BeginDate
  ElapsedTime = ( (EndTime - 1) + (ElapsedDays * 8640000) ) |
                - (BeginTime - 1) + 1
End
```

In fact, if you can deal with four variables, I think this would handle all your ET computing needs.

StarDates

StarDates are the most often mentioned solution to the problem of cross-date time computations. Some are not comfortable with tracking four variables, as above. A StarDate needs only two (and one can be a local variable).

I have it on reliable authority that the StarDate technique was introduced in Dave Howington's Logix Models for Clarion Professional Developer (therefore, about 15 years ago). It is exceedingly ingenious. A StarDate uses a single variable, typically a `Real(15,8)` or `Decimal (13,6)` – without entering into the Real vs. Decimal debate, both these data type, pictured as shown, do work – to contain both the date and the time. Therefore, two times can be compared using StarDates without worrying about midnight rollovers.

A StarDate is constructed by taking the Clarion standard date and adding the time expressed as a fraction. Thus, the integer portion of a StarDate is the date and the fractional portion is the time.

Time can be expressed as a fraction by dividing it by 8640000 (the maximum number of clock ticks in a day). For example:

```
    BeginDate = Today() + ( Clock() / 8640000 )
```

StarDates are converted back into a date and a time by reversing the arithmetic.

```
    BeginDate = Int(StarDate)
```

The time is the fractional portion and I get it by subtracting the date:

```
    BeginTime = StarDate - INT(StarDate)
```

To finish the conversion to a Clarion standard time, multiply by 8640000.

```
    BeginTime *= 8640000
```

So,

```
    If Today() > INT(Timeout)
```

there has been a midnight rollover and I can check `EndTime + 8640000`, as above.

Summary

Personally, I find that StarDates are great when testing whether a time out period has expired (see Replicating IDLE: All Quiet on the Keyboard?). When the dates and times are already in a file or queue, the codelet above does the job.

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

Reader Comments

Add a comment

- » Hi Steve, In the past years I've done some time...
- » Actually, my interest is in computing elapsed time. For...

Clarion Magazine

Improving On The Non-Related Lookup Template

by Nardus Swanevelder

Published 2006-07-17

Clarion ships with a template called Lookup Non-Related Record. This template is a code template and it allows you to do a lookup from a table that is not related to the primary table. While the template handles basic lookups just fine, it doesn't always do everything I want it to do. And while I could write embed code for the additional functionality, a better long-term solution is to extend the template, which is what I will demonstrate in this article.

Using the stock template

Let's say you have a `Lookup` table that consists of a system ID and a description, and you want to display the description based on the system ID stored in another table. You can do this with the Lookup Non-Related Record template.

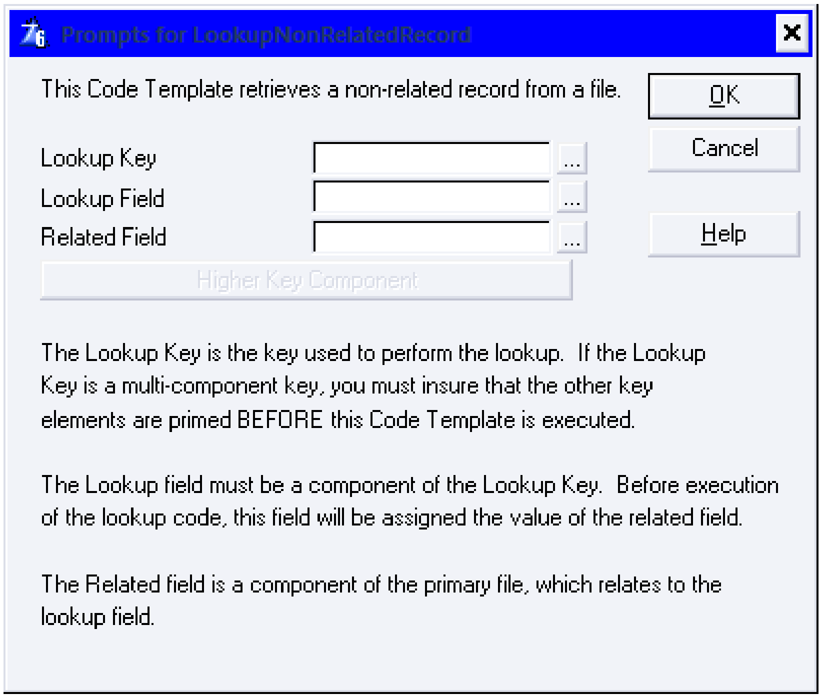
Here's an example using a tax lookup:

```
Tax Lookup Table - Pre(TAX)


PK_Tax                Key(TAX:SysIDTax)                  !Primary Key
SysIdTax             Long
DescriptionTax       String(50)



Client Table - Pre(CLI)


SysIdClient          Long
FirstNameClient      String(50)
SurnameClient        String(50)
SysIdTax             Long
```

In your update form for the client you won't display the `SysIdTax` value; you would rather display the Tax Description. One way to do this is to populate the `TAX:DescriptionTax` field on your update screen, but the problem is that you have to make sure that the correct tax record is active when you open the update form.The code to do this will be something like this:

```
TAX:SysIdTax = CLI:SysIdTax
Access:Tax.Fetch(TAX:PK_Tax)
```

The Lookup Non-Related Record template generates this code exactly. But why use this template, since it is not a lot of code to type? Two reasons. The first reason is that it is easier to just click on buttons and select values compared to trying to remember the exact spelling of the field names, queue names etc. The second reason is that if you make a change to the field name in the dictionary Clarion will automatically change the code for you, so no compile errors.



This Code Template retrieves a non-related record from a file.

The Lookup Key is the key used to perform the lookup. If the Lookup Key is a multi-component key, you must insure that the other key elements are primed BEFORE this Code Template is executed.

The Lookup field must be a component of the Lookup Key. Before execution of the lookup code, this field will be assigned the value of the related field.

The Related field is a component of the primary file, which relates to the lookup field.

Figure 1. Clarion Lookup Non Related Record

I use the scenario above quite a lot but I needed some added functionality. One of the things I do differently is that I don't populate the `TAX:DescriptionTax` field onto the update form, I populate a local variable onto the update form that I derive from the `TAX:DescriptionTax` field. How does this change the code mentioned above?

```
TAX:SysIdTax = CLI:SysIdTax
Access:Tax.Fetch(TAX:PK_Tax)
LOC:DescriptionTax = TAX:DescriptionTax
```

The other problem I have with the Clarion template is that it will do the fetch even if the user is inserting a record. This is not acceptable to me; instead, this is what I want my code to look like:

```
If Self.Request <> ChangeRecord
   TAX:SysIdTax = CLI:SysIdTax
   Access:Tax.Fetch(TAX:PK_Tax)
   LOC:DescriptionTax = TAX:DescriptionTax
End
```

So why create a new template? Why not just add the extra code around Clarion's template code? Firstly I am lazy; I don't want to have to type all that code if I don't have to. Secondly, if I change my field names it would be nice if Clarion changes my code for me, and that will only happen if the code is template driven.

As always when I am creating a new template I first look for a template that does more or less what I need. Clarion supplied as with the Non-Related Lookup template so this was my starting point.

You can find the Clarion code in abcode.tpw in your Clarion template directory. (You can also find the code in code.tpw but this is the legacy code) The Clarion template has a lot of code that I don't really need, particularly related to multi-part keys (some line breaks added for readability):

```
#!---------------------------------------------------------
#CODE(LookupNonRelatedRecord,'Lookup Non-Related Record'),
  HLP('~TPLCodeLookupNonRelatedRecord')
#DISPLAY('This Code Template retrieves a non-related record from a file.')
#DISPLAY('')
#PREPARE
   #CALL(%LoadHigherKeysComponets,%FindFileFromKey(%LookupKey),
     %LookupKey,%LookupField,'Single Value')
#ENDPREPARE
#PROMPT('Lookup Key',KEY),%LookupKey,REQ,WHENACCEPTED(%LoadHigherKeysComponets(
  %FindFileFromKey(%LookupKey),%LookupKey,%LookupField,'Single Value'))
#PROMPT('Lookup Field',COMPONENT(%LookupKey)),%LookupField,REQ,WHENACCEPTED(
  %LoadHigherKeysComponets(%FindFileFromKey(%LookupKey),%LookupKey
  ,%LookupField,'Single Value'))
#PROMPT('Related Field',FIELD),%RelatedField,REQ
#BOXED,WHERE(%False),AT(0,0,0,0)
   #BUTTON('LookupHigherKeys'),MULTI(%HigherKeys,%HigherKey)
      #PROMPT('Lookup Higher Field',COMPONENT(%LookupKey)),%HigherKey
   #ENDBUTTON
#ENDBOXED
#ENABLE(ITEMS(%HigherKeys))
#BUTTON('Higher Key Component'),FROM(%HigherKeys,%HigherKeyField&' = '&
  CHOOSE(%HigherKeyValueType,'Current Value',%HigherKeyValueVariable,
  %HigherKeyValueFixed)),AT(,,175),HLP('~TPLHigherKeyComponent')
   #ENABLE(%False)
      #PROMPT ('Key Field:', FIELD),%HigherKeyField,DEFAULT(%HigherKey)
   #ENDENABLE
   #PROMPT('Value Type:',DROP('Current Value[1]|Variable Value[2]|
     Fixed Value[3]')),%HigherKeyValueType,DEFAULT(1)
   #BOXED,WHERE(%HigherKeyValueType=2),AT(,30)
      #PROMPT ('Value:', FIELD),%HigherKeyValueVariable
   #ENDBOXED
   #BOXED,WHERE(%HigherKeyValueType=3),AT(,30)
      #PROMPT ('Value:', EXPR),%HigherKeyValueFixed
   #ENDBOXED
#ENDBUTTON
#ENDENABLE
#DISPLAY('')
#DISPLAY('The Lookup Key is the key used to perform the lookup.  If the Lookup')
#DISPLAY('Key is a multi-component key, you must insure that the other key')
#DISPLAY('elements are primed BEFORE this Code Template is executed.')
#DISPLAY('')
#DISPLAY('The Lookup field must be a component of the Lookup Key.  Before execution')
#DISPLAY('of the lookup code, this field will be assigned the value of the related
field.')
#DISPLAY('')
#DISPLAY('The Related field is a component of the primary file, which relates to
the')
#DISPLAY('lookup field.')
#DISPLAY('')
#IF(ITEMS(%HigherKeys))
  #FOR(%HigherKeys),WHERE(%HigherKeyValueType<>1)
#IF(%HigherKeyValueType=2)
 %HigherKeyField = %HigherKeyValueVariable
#ELSE
 %HigherKeyField = %HigherKeyValueFixed
#ENDIF
```

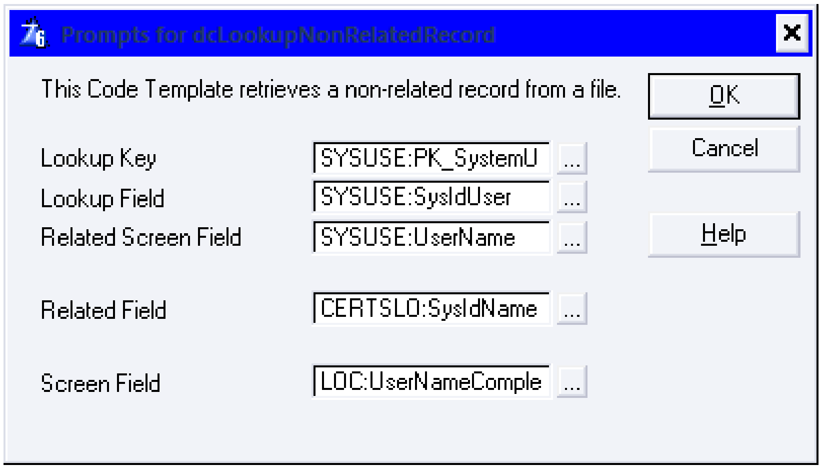
```
  #ENDFOR
#ENDIF
 %LookupField = %RelatedField      #<! Move value for lookup
#FIND(%Field,%LookupField)         #!  FIX field for lookup
 Access:%File.Fetch(%LookupKey)    #<! Get value from file
#ATSTART
#CALL(%LoadHigherKeysComponets,%FindFileFromKey(%LookupKey),
  %LookupKey,%LookupField,'Single Value')
#ENDAT
#!---------------------------------------------------------
#GROUP(%FindFileFromKey,%parKey),PRESERVE
#DECLARE(%MyFile)
#FOR(%File)
  #FOR(%Key),WHERE(%Key=%parKey)
     #RETURN(%File)
  #ENDFOR
#ENDFOR
#RETURN('')
```

I removed all the code I don’t need and added the necessary code to do the assignment I want; I also added the code to check the `Self.Request` status.

This is how my template code looks:

```
#CODE(dcLookupNonRelatedRecord,'DC Lookup Non-Related Record')
#DISPLAY ('This Code Template retrieves a non-related record from a file.')
#DISPLAY ('')
#PROMPT  ('Lookup Key',KEY),%LookupKey,REQ
#PROMPT  ('Lookup Field',COMPONENT(%LookupKey)),%LookupField,REQ
#PROMPT  ('Related Screen Field',FIELD),%RelatedScreenField,REQ
#DISPLAY ('')
#PROMPT  ('Related Field',FIELD),%RelatedField,REQ
#DISPLAY ('')
#PROMPT  ('Screen Field',FIELD),%ScreenField,REQ
#DISPLAY ('')
If Self.Request <> InsertRecord
   %LookupField = %RelatedField      #<! Move value for lookup
#FIND(%Field,%LookupField)           #!  FIX field for lookup
   Access:%File.Fetch(%LookupKey)    #<! Get value from file
   %ScreenField = %RelatedScreenField
End
```

Nice and short – see Figure 2 for an example of the template prompts.



Prompts for dcLookupNonRelatedRecord

This Code Template retrieves a non-related record from a file.

Lookup Key SYSUSE:PK_SystemU

Lookup Field SYSUSE:SysIdUser

Related Screen Field SYSUSE:UserName

Related Field CERTSLO:SysIdName

Screen Field LOC:UserNameComple

Figure 2. dc Lookup Non Related Record

The above template will generate the following code:

```
If Self.Request <> InsertRecord
   SYSUSE:SysIdUser = CERTSLO:SysIdNameCompletedForm
   Access:SystemUsers.Fetch(SYSUSE:PK_SystemUsers)
   LOC:UserNameCompletedForm = SYSUSE:UserName
End
```

Summary

By simplifying the existing Clarion Lookup Non-Related Record template, I was able to retain all the functionality I need for my lookups, while easily adding a few new features, including an additional Screen Field prompt and conditional lookup code execution.

Download the source

Nardus Swanevelder was born and raised in South Africa. He was a networking engineer for seven years before he moved over to the commercial side of the business. Nardus has developed a Sale Cycle Management system for the Information and Communication Technology industry. He has been programming in Clarion since 1989, and holds B.Com and MBA degrees. In his spare time Nardus lectures Financial Management to B. Com Hons students at North-West University.

Reader Comments

Add a comment

Clarion Magazine

Beautifying Clarion Applications

by Roel Abspoel

Published 2006-07-13

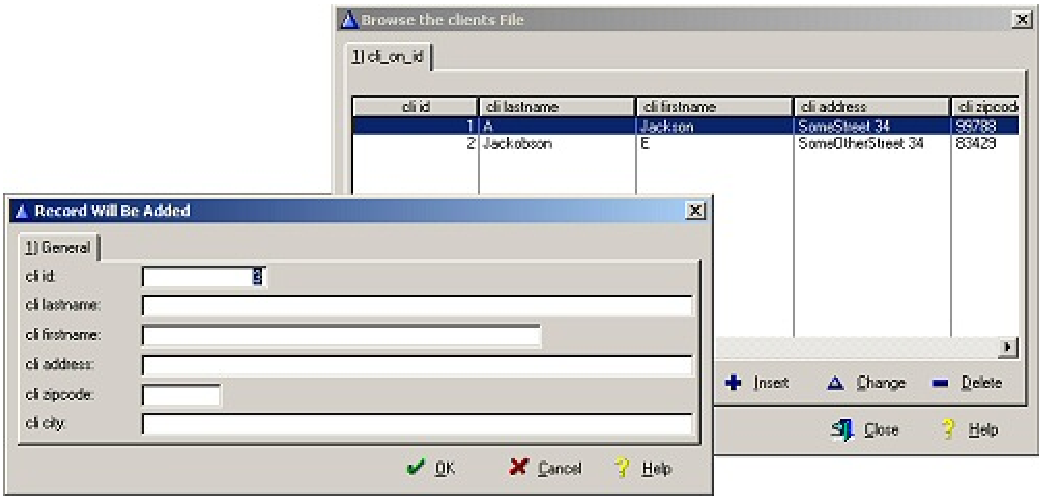
In this article I will show you a simple way to make your application more appealing to the eye. Now, I can just hear you all saying "Beauty is only skin deep" or in this case GUI deep. Consider this: You may have spend countless hours perfecting the code in your application and forming great functionality based on your extensive experience in a particular field. Maybe your application is unique in features and just maybe no other developer can offer the functionality that you can. But does your application’s appearance represent the genius coding underneath or is your application just an old rusty noisy car with a supercharged engine?

Now don’t start blaming Clarion; your application is a child sprouted from both you and Clarion. Mother Clarion may be old and ugly but it still offers you a world of options, and all the genes are in place and ready for use. The mother isn’t the problem in this case, it’s the seeds from the father (or if these words are starting to make you feel uncomfortable, "the input from the coder") that is the problem here.

Highlighting prompts and lists

Why settle for Clarion generated browses on an outdated tab control? Why have your prompts in a form placed on that same tab control?

It doesn’t make sense to me to place either on a tab, especially now that Clarion offers sorting by clicking on list headers. Tabs clutter your screen and frankly confuse the end-user.



cli id	cli lastname	cli firstname	cli address	cli zipcod
1	A	Jackson	SomeStreet 34	99788
2	Jackobson	E	SomeOtherStreet 34	83429

cli id: 3
cli lastname:
cli firstname:
cli address:
cli zipcode:
cli city:

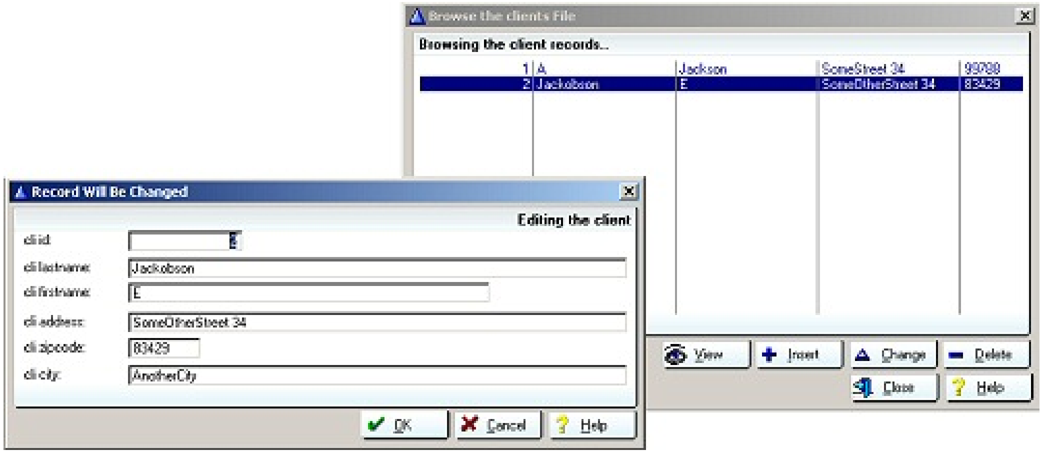
Figure 1. Standard Clarion generated browse and update (view full size image)

There is an argument that box-shaped controls can indicate to an end-user that this list is important or these prompts are some kind of group that need to be seen as a whole. A tab or transparent panel, though, confuses your end-users because although it is beveled it doesn't really distinguish your controls from the background.

What controls are alternatives to the tab to make your prompts and lists look more attractive? There are panel and box controls, but let's be honest, they're not much to look at either. Adding a graphic underneath your prompts and lists doesn't work either when your end-user's display size is different from the one you used at design time.

So that's it then? No 'out of the box' solutions for this display problem?

Well, remember I said it was "all your" fault? Well it is! Why is it we're so creative when it comes to coding features yet so limited when it comes to the user interface? Let's start with a more creative approach; how do you *want* your application to look? For me that meant I did some research with Google's picture-search. When I couldn't find an example that pleased me I realized that my browser's start page was showing me just the thing I had in mind: Microsoft's Live.com website featured nice glassy looking panels for each separate item. I could just see my app with these kind of panels lighting up my application.



Browse the clients File

Browsing the client records..

1	A	Jackson	SomeStreet 34	99788
2	Jackobson	E	SomeOtherStreet 34	83429

Record Will Be Changed

Editing the client

cli:id: E
cli:lastname: Jackobson
cli:firstname: E
cli:address: SomeOtherStreet 34
cli:zipcode: 83429
cli:city: AnotherCity

Figure 2. Dreaming of glass panels and buttons (view full size image)

But of course I now needed a way to produce this effect in Clarion, and preferably without too much effort.

Redrawing controls

What I'm going to do now is create a procedure that lets me redraw a control. The procedure should take a control as a parameter, and some bytes to indicate if It should draw a header, shadow and if the control is a button or not (for buttons I'm going to change some properties.)

If I'm going to draw a new panel on the window instead of the old one (be it a panel, button or box or whatever) I'm going to have to know what to draw and with what dimensions. So lets zoom in on the glass panel. Drawing this panel is done in four phases:

1. The panel surroundings, and possibly a drop shadow.
2. The panel header with a gradient
3. The panel body which is just a bunch of white lines.
4. The panel footer which again is a gradient.

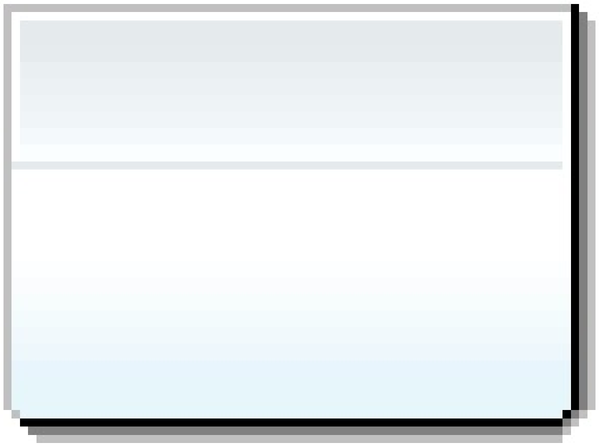


Figure 3. Zooming in on the glass panel to see how it is build up

Almost all of this panel can be done by making horizontal lines in a certain color. The only exceptions are the panel surroundings and the shadow lines which are only 1 pixel in width. These are drawn obviously with vertical lines. Maybe you noticed that the panel is square on the header but rounded on the bottom; this also means I have to connect my horizontal and vertical surrounding lines with a diagonal and make my footer lines a bit smaller than the body lines.

The DrawGlassPanel procedure

All of the drawing is done by a single procedure called `DrawGlassPanel`. This procedure is contained in the template, which is included in the downloadable source at the end of this article.

When drawing graphics I prefer to use pixels instead of dialog units (which are the default screen measurement units.). For this I need to instruct the window to measure in pixels.

I don't want to mess up other template or manual code so I need to make sure that I turn this on only for my procedure, and after it is completed I switch back to the original measurement units. I define a byte and store the current value of `PROP:PIXELS` (a byte, `True` or `False`), then set the pixels property to `True`. At the end of the procedure I set `PROP:PIXELS` back to the original state.

```
DrawGlassPanel  PROCEDURE(LONG MyControl,BYTE DrawHeader,|
                             BYTE DrawShadow,<BYTE IsButton>)
OrgPixels             = 0{PROP:PIXELS}
0{PROP:PIXELS} = TRUE
! Drawing code
0{PROP:PIXELS} = OrgPixels
```

Maybe you noticed that I have defined a parameter called `IsButton`. This parameter is to tell the procedure that the control I specified is in fact a button. The reason for this is that in case of a button I need to set some properties that I would not set for a panel or box. So my next step in this procedure is to set the properties for my defined control.

```
IF IsButton = TRUE
  MyControl{PROP:FLAT} = TRUE
  MyControl{PROP:SKIP} = TRUE
  IF ~MyControl{PROP:TIP}
    MyControl{PROP:TIP}  = MyControl{PROP:TEXT}
  END
  HiddenButton = MyControl{PROP:HIDE}
```

```
    ELSE
      MyControl{PROP:HIDE} = TRUE
    END
```

This code hides all non-button controls and makes button controls flat (I'm drawing my own bevel). Also the button is set to be skipped in the tab order but if you prefer then you can comment this out. If your button doesn't have a tooltip the procedure sets it to the button text and the variable `HiddenButton` stores the button's hide state; if it is hidden I don't want to draw a new button.

Drawing the line(s)

The first thing is to draw the surrounding lines for the control. These lines can be drawn in two ways with shadow and without. If the parameter `DrawShadow` is set to true then the surrounding lines on the top and left will be gray and the lines on the bottom and right will be black (the shadow will drop from the right and bottom so I start the shadow from black; if you wanted the shadow the other way around then also change the line colors here.)

Drawing the lines is pretty easy; the Clarion `LINE` procedure will take care of this. But be sure to start by setting the line thickness and color. `SetPenWidth(1)` will set the thickness to one pixel which is exactly what I want. `SetPenColor()` will set the color of the line I'm about to draw.

The hard part is to determine at what `X` and `Y` position to draw and how long and in what direction.

```
    IF DrawShadow = TRUE
      SetPenColor(COLOR:BLACK)
    ELSE
      SetPenColor(13883352)
    END
    LINE((MyControl{PROP:XPOS}+MyControl{PROP:WIDTH})|
      ,MyControl{PROP:YPOS},0,(MyControl{PROP:HEIGHT}-2))
    LINE((MyControl{PROP:XPOS}+MyControl{PROP:WIDTH}),|
      (MyControl{PROP:YPOS}+(MyControl{PROP:HEIGHT}-2)),-2,2)
    LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}|
       +MyControl{PROP:HEIGHT}),(MyControl{PROP:WIDTH}-3),0)
    IF DrawShadow = TRUE
      SetPenColor(COLOR:SILVER)
    ELSE
      SetPenColor(13883352)
    END
    LINE(MyControl{PROP:XPOS},MyControl{PROP:YPOS},|
      MyControl{PROP:WIDTH},0)
    LINE(MyControl{PROP:XPOS},MyControl{PROP:YPOS},0,|
      (MyControl{PROP:HEIGHT}-2))
    LINE(MyControl{PROP:XPOS},(MyControl{PROP:YPOS}+|
      (MyControl{PROP:HEIGHT}-2)),3,3)
```

Note that the pen color I'm setting depends on the `DrawShadow` parameter. This is because if I don't want a shadow underneath my panel I want the panel to sink into the background. I could just not draw a border at all in this case, but this makes the panel kind of bleed into my window. So if there should not be a shadow, a very light gray line is drawn around the panel, lighter than the default `COLOR:SILVER`.

If you noticed, each horizontal and vertical side (up and right, down and left) consists of three lines, one horizontal, one vertical and one diagonal. The `X` and `Y` are based upon the location off the original control and the length of the line is determined by the width or height of my original control.

Dropping a shadow

The drop shadow consists of 2 lines. (but is started after the black right and bottom line.)

```
IF DrawShadow = TRUE
  SetPenColor(COLOR:GRAY)
  LINE((MyControl{PROP:XPOS}+MyControl{PROP:WIDTH})+1,|
    (MyControl{PROP:YPOS}+1),0,(MyControl{PROP:HEIGHT})-2)
  LINE((MyControl{PROP:XPOS}+MyControl{PROP:WIDTH})+1,|
    (MyControl{PROP:YPOS}+MyControl{PROP:HEIGHT})-2,-3,3)
  LINE((MyControl{PROP:XPOS}+3),(MyControl{PROP:YPOS}+|
    MyControl{PROP:HEIGHT})+1,(MyControl{PROP:WIDTH}-4),0)
  SetPenColor(COLOR:SILVER)
  LINE((MyControl{PROP:XPOS}+MyControl{PROP:WIDTH})+2,|
    (MyControl{PROP:YPOS}+2),0,(MyControl{PROP:HEIGHT})-3)
  LINE((MyControl{PROP:XPOS}+MyControl{PROP:WIDTH})+2,|
    (MyControl{PROP:YPOS}+MyControl{PROP:HEIGHT})-2,-4,4)
  LINE((MyControl{PROP:XPOS}+4),(MyControl{PROP:YPOS}+|
    MyControl{PROP:HEIGHT})+2,(MyControl{PROP:WIDTH}-5),0)
END
```

Basically this is a black line, followed by a gray line, followed by a silver line.

Again each shadow line is made up out of three lines.

Getting a head(er)

If the parameter DrawHeader is set to true it means I need to draw a header on the panel.

The header has a total height of 17 pixels (*plus* the surrounding lines to make the header stand out more) boxing your header in. The gradient covers 10 colors in total. (If you want your panel to have a different color you should change these 10 values to new gradient values; you could simply create a nice gradient with your favorite graphics program and select 10 colors from the image.) Each line is drawn, and for each gradient change a new color is set using `SetPenColor`. The `Y` position increments for every line:

```
IF DrawHeader = TRUE
  SetPenColor(16777215)
  LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+1),|
    (MyControl{PROP:WIDTH}-2),0)
  LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+1),0,18)
  LINE((MyControl{PROP:XPOS}+MyControl{PROP:WIDTH})-1,|
    (MyControl{PROP:YPOS}+1),0,18)
  SetPenColor(15592421)
  LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+2),|
    (MyControl{PROP:WIDTH}-3),0)
  LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+3),|
    (MyControl{PROP:WIDTH}-3),0)
  LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+4),|
    (MyControl{PROP:WIDTH}-3),0)
```

```
      SetPenColor(15658214)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+5),|
        (MyControl{PROP:WIDTH}-3),0)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+6),|
        (MyControl{PROP:WIDTH}-3),0)
      SetPenColor(15921386)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+7),|
        (MyControl{PROP:WIDTH}-3),0)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+8),|
        (MyControl{PROP:WIDTH}-3),0)
      SetPenColor(16052972)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+9),|
        (MyControl{PROP:WIDTH}-3),0)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+10),|
        (MyControl{PROP:WIDTH}-3),0)
      SetPenColor(16184558)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+11),|
        (MyControl{PROP:WIDTH}-3),0)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+12),|
        (MyControl{PROP:WIDTH}-3),0)
      SetPenColor(16316144)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+13),|
        (MyControl{PROP:WIDTH}-3),0)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+14),|
        (MyControl{PROP:WIDTH}-3),0)
      SetPenColor(16579316)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+15),|
        (MyControl{PROP:WIDTH}-3),0)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+16),|
        (MyControl{PROP:WIDTH}-3),0)
      SetPenColor(16777210)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+17),|
        (MyControl{PROP:WIDTH}-3),0)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+18),|
        (MyControl{PROP:WIDTH}-3),0)
      SetPenColor(15592421)
      LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+19),|
        (MyControl{PROP:WIDTH}-3),0)
      CurPos = 20
    ELSE
      CurPos = 1
    END
```

When the header is drawn the variable `CurPos` is set to a value. This is the `Y` position where I will start when I draw the panel body.

Adding to your body count

Now that I have a header (or not, depending on the `DrawHeader` parameter) I can start by drawing the panel body. I could skip this part entirely which would make the panel transparent by showing the window background color, however this may not be such a great idea. The gradient I have chosen depends on a white background so if the window is silver it would make one silly looking panel.

```
    SetPenColor(16777215)
```

```
LOOP cnt#=CURPOS TO (MyControl{PROP:HEIGHT}-20)
  LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+cnt#),(MyControl{PROP:WIDTH}-
1),0)
END
```

Since I’m only using one color for the panel body I can simply loop the lines. If you want to use a gradient here you’ll need to add a bit more code.

In the header I decided what the starting position would be by setting `CurPos`.

So I’m going to loop starting from `CurPos` until my position is equal to my original panel’s height minus 20 (pixel height for the footer).

The end of the line

To finish things off I still need to create a footer.

This is done basically the same as creating the header, with some exceptions:

1. Unlike the header this gradient doesn’t need to be boxed in. Instead, the footer gradient should be as wide as the body.
2. The bottom of the panel is rounded which means the x position of the last two lines should be incremented one pixel each. As a result, the width should be decreased to match the rounding on the right.

The gradient is chopped up in eight parts, each part with their own color. The last part consists of five lines all in the same color; this emphasizes the gradient and ‘thicknesses of the panel more.

```
SetPenColor(16776957)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+0))|
  ,(MyControl{PROP:WIDTH}-1),0)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+1)),|
  (MyControl{PROP:WIDTH}-1),0)
SetPenColor(16711162)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+2)),|
  (MyControl{PROP:WIDTH}-1),0)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+3)),|
  (MyControl{PROP:WIDTH}-1),0)
SetPenColor(16645367)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+4)),|
  (MyControl{PROP:WIDTH}-1),0)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+5)),|
  (MyControl{PROP:WIDTH}-1),0)
SetPenColor(16644852)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+6)),|
  (MyControl{PROP:WIDTH}-1),0)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+7)),|
  (MyControl{PROP:WIDTH}-1),0)
SetPenColor(16579057)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+8)),|
  (MyControl{PROP:WIDTH}-1),0)
LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+9)),|
  (MyControl{PROP:WIDTH}-1),0)
```

```
    SetPenColor(16578798)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+10)),|
      (MyControl{PROP:WIDTH}-1),0)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+11)),|
      (MyControl{PROP:WIDTH}-1),0)
    SetPenColor(16578280)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+12)),|
      (MyControl{PROP:WIDTH}-1),0)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+13)),|
      (MyControl{PROP:WIDTH}-1),0)
    SetPenColor(16447207)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+14)),|
      (MyControl{PROP:WIDTH}-1),0)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+15)),|
      (MyControl{PROP:WIDTH}-1),0)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+16)),|
      (MyControl{PROP:WIDTH}-1),0)
    LINE((MyControl{PROP:XPOS}+1),(MyControl{PROP:YPOS}+(cnt#+17)),|
      (MyControl{PROP:WIDTH}-1),0)
    LINE((MyControl{PROP:XPOS}+2),(MyControl{PROP:YPOS}+(cnt#+18)),|
      (MyControl{PROP:WIDTH}-3),0)
```

Make up

Now that the procedure is finished I can add it to my app as a global procedure. Then from any window I can alter the panels and buttons and whatever else I want.

So that’s it, I’m done with the makeup, and I can now add lovely glass panels to my application!

What’s that? You’re not still mad at me because I said ‘it was all your fault’, are you? You are? Well okay then, I’ll make it up to you by taking this a few steps further.

The template

If you start changing your buttons and panels, you will agree with me it is a bit of a hassle to do this for every button/panel you have. And what about resizing controls at runtime?

So here’s a thought, why not put all of this configuration information into a template?

If you download the example application you will also get a nice .tpl file with all the mentioned code.

Add the global extension to add the `DrawGlassPanel` procedure to your application, and add the procedure extension locally to a window to configure what controls you want to upgrade.

The template provides two tabs:

Glass panels

This is where you can manually select which controls you want to turn into glass panels.

Choose if you want a header on the panel and if you want a drop shadow.

Glass buttons

This tab is default set to ‘Do not update any buttons’.

Remove this check to update all the buttons on your window.

You can choose if the buttons should have a drop shadow or not.

If you like, you can exclude certain buttons by simply adding them to the exclude list.

If you remove the check ‘Update all buttons on the window’ you can now add the buttons you do want changed to the ‘Upgrade only these buttons’ list. In this case you can choose for each button if you want a drop shadow or not.

Important note: If you are using the template in Clarion versions lower than 6 then the `BLANK` function used in the window resize doesn’t exist (it was introduced in C6). This means that in C5.5 or lower resizing the window will not resize the panel.

Conclusion

I hope to have demonstrated that a little time and creativity can make a lot of difference in your application. Sure beauty is only skin or GUI deep, but that doesn’t mean it has no role in selling your app. An application that looks good and works according to standards will sell much easier than ugly or awkward-to-use software. Think about how you want your app to look and behave, and find a way to make it happen! There are lots of freeware additions available and even more high quality commercial options to upgrade your application. Invest some time and creativity, or if you lack either of these, some money, to give your application the presentation it deserves!

Download the source

Reader Comments

Add a comment

Clarion Magazine

Another Single Browse For Multiple Lookups

by Nardus Swanevelder

Published 2006-07-10

I recently started working on a new project where I had to code a browse and an update screen for 22 tables that only of only two fields each. The two fields in the table are an autonumber field, declared as a `LONG`, and a description field, declared as a `STRING`. The coding of 22 similar browse and update screens seemed like a lot of duplication, and that is when I recalled the article A Single Browse For Multiple Lookups, was written by Randy Rogers in 2002.

In his article Randy wrote: "I am sure most Clarion programmers have been faced with the same dilemma. After normalizing your database you find you have many tables that are basically the same: a code and a description."

His plan to fix this problem was simple, and I quote: "compare two browses to discover the differences, then create a common browse procedure that would receive the differences as parameters and use edit-in-place to update the fields. Once the common browse was completed, I intended to create a template that would generate the code to call the common browse for each of the lookup tables."

The only issue I had with this approach was the way the template made the decision to generate the code to call the common browse. His argument that "I knew all the non-lookup tables in the dictionary had more than two fields, so I created the template to generate code for all files with only two fields" is valid, but my experience showed me that sometimes I didn’t want to include all the two field tables in the common lookup template.

One solution Randy offered was to use the User Option for the tables that will call the common browse, but he felt that would be too much work. I agree that it is a lot of work adding the user option to all the tables but the biggest problem I have with this approach is that you need to go into the data dictionary and look at individual tables to see which tables will call the common browse.

The approach that I took does mean that you have to do some typing up front, but after that the process to add or remove lookup tables from and to your application is quite easy. I will cover this in more detail later on.

Due to the fact that Randy had written his article nearly four years ago I also decided to re look at the code for the common browse.

Before I explain the browse code, I need to highlight the fact that my solution only works for a multi-DLL application. That is you need at least a data DLL and the main EXE. The source procedure will be added to the data DLL and the generated procedures will be called from the main EXE.

The browse code

Okay, back to the code, I started by comparing two of the browses I have generated. What I found was that each browse required a caption, a view, a relation manager, two keys, a pointer to the autonumber field, a pointer to the description field and the procedure name. Armed with this list of differences I took one of the generated browses and used it as the starting point to code a source procedure. When creating the procedure prototype, I added a prefix of "sz" to the labels that had been created by the generator; this was to minimize the code changes I would need to make. Here’s what the final prototype looks like:

```
GenerateLookupBrowse PROCEDURE  (*cstring szCaption, |
  VIEW BRW1::View:Browse, RelationManager szRM, KEY szKeyN, |
  KEY szKeyA,*? szAutoNumber, *? szDescription, *cstring szProcName)
```

Every browse has a `Queue` that contains `LIKE` variables for each of the table fields. Since all of the lookups will have the same format, I modified the generated `Queue` to reflect this known data layout.

Queue - before

```
Queue:Browse:1                     QUEUE
LSTAPHOTYP:SysIdStaffPhotoType       LIKE(LSTAPHOTYP:SysIdStaffPhotoType)
LSTAPHOTYP:DescriptionStaffPhotoType LIKE(LSTAPHOTYP:DescriptionStaffPhotoType)
LSTAPHOTYP:CounterPrefix                     LIKE(LSTAPHOTYP:CounterPrefix)
Mark                                         BYTE
ViewPosition                                 STRING(1024)
                                   END
```

Queue - after

```
Queue:Browse:1       QUEUE
AutoNumber              LONG
Description             CString(512)
Mark                    BYTE
ViewPosition            STRING(1024)
                     END
```

Only two of the methods need to be changed, and those lines are in boldface.

```
ThisWindow.Init
GlobalErrors.SetProcedureName(szProcName)
SELF.Request = GlobalRequest
ReturnValue = PARENT.Init()
IF ReturnValue THEN RETURN ReturnValue.
SELF.FirstField = ?Browse:1
SELF.VCRRequest &= VCRRequest
SELF.Errors &= GlobalErrors
SELF.AddItem(Toolbar)
CLEAR(GlobalRequest)
CLEAR(GlobalResponse)
IF SELF.Request = SelectRecord
   SELF.AddItem(?Close,RequestCancelled)
ELSE
   SELF.AddItem(?Close,RequestCompleted)
END
szRM.Open
SELF.FilesOpened = True
BRW1.Init(?Browse:1,Queue:Browse:1.ViewPosition,|
  BRW1::View:Browse,Queue:Browse:1,szRM,SELF)
SELF.Open(QuickWindow)
```

```
QuickWindow{Prop:Text} = szCaption
Do DefineListboxStyle
BRW1.Q &= Queue:Browse:1
BRW1::Sort1:StepClass.Init(+ScrollSort:AllowAlpha,ScrollBy:Runtime)
BRW1.AddSortOrder(BRW1::Sort1:StepClass,szKeyA)
BRW1.AddLocator(BRW1::Sort1:Locator)
BRW1::Sort1:Locator.Init(,szDescription,1,BRW1)
BRW1::Sort0:StepClass.Init(+ScrollSort:AllowAlpha)
BRW1.AddSortOrder(BRW1::Sort0:StepClass,szKeyN)
BRW1.AddLocator(BRW1::Sort0:Locator)
BRW1::Sort0:Locator.Init(,szAutoNumber,1,BRW1)
BRW1.AddField(szAutoNumber,BRW1.Q.AutoNumber)
BRW1.AddField(szDescription,BRW1.Q.Description)
Resizer.Init(AppStrategy:Surface,Resize:SetMinSize)
SELF.AddItem(Resizer)
INIMgr.Fetch(szProcName,QuickWindow)
Resizer.Resize
QuickWindow{prop:buffer} = 1
BRW1.AddToolbarTarget(Toolbar)
BRW1.ToolbarItem.HelpButton = ?Help
SELF.SetAlerts()
RETURN ReturnValue
ThisWindow.Kill
IF SELF.FilesOpened
   szRM.Close
END
IF SELF.Opened
   INIMgr.Update(szProcName,QuickWindow)
END
```

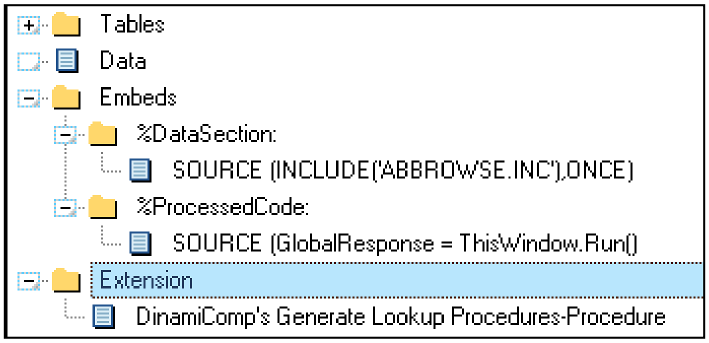
The extension template

Here’s the code for the extension template that converts a regular browse to a reusable browse:

```
#!---------------------------------------------------------
#TEMPLATE(dcLookups, 'DinamiComp - Common Lookup Generator'),FAMILY('ABC')
#!---------------------------------------------------------
#!
#!
#SYSTEM
#DECLARE (%UModule)
#!
#!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
#!            PROCEDURE EXTENSION
#!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
#!
#EXTENSION(dcGenerateLookups,'DinamiComp''s Generate Lookup Procedures-Procedure')
#!
#SHEET
#TAB('General')
    #Insert(%SETCC)
    #BOXED   ('Debugging'),Section,AT(,95,,20)
      #PROMPT  ('Disable All DinamiComp Templates',Check),%DisableAll,AT(10)
    #ENDBOXED
#ENDTAB
```

```
#TAB('Settings')
    #Insert(%SETAPP)
    #DISPLAY ('')
    #BOXED   ('Options'),AT(,65,,40)
       #DISPLAY ('')
       #DISPLAY ('Source Lookup Procedure Name')
       #PROMPT  ('',@s150),%SourceLookupProcName,AT(,,180,)
    #ENDBOXED
#ENDTAB
#TAB('Files')
    #DISPLAY ('')
    #BUTTON     ('Lookup Files...'),MULTI(%LookupFileMultiCount,
                 %LookupFileMulti & ' : ' & %ProcedureNameMulti
                         & ' : ' & %ScreenCaptionMulti),INLINE
       #DISPLAY ('Screen Caption')
       #PROMPT  ('',@s255),%ScreenCaptionMulti,REQ,DEFAULT(''),At(,,180,)
       #DISPLAY ('Select Lookup File.')
       #PROMPT  ('',FILE),%LookupFileMulti,REQ,
         WHENACCEPTED(%StripExclamation(%LookupFileMulti))
       #DISPLAY ('Select lookup numeric key')
       #PROMPT  ('',KEY),%LookupNumericKeyMulti,REQ,AT(,,150,)
       #DISPLAY ('Select lookup alphabetic key')
       #PROMPT  ('',KEY),%LookupAlphabeticKeyMulti,REQ,AT(,,150,)
       #DISPLAY ('Select Lookup field 1 (AutoNumber)')
       #PROMPT  ('',FIELD),%LookupFieldIDMulti1,REQ,
             WHENACCEPTED(%StripExclamation(%LookupFieldIDMulti1))
       #DISPLAY ('Select Lookup field 2 (Description)')
       #PROMPT  ('',FIELD),%LookupFieldIDMulti2,REQ,
             WHENACCEPTED(%StripExclamation(%LookupFieldIDMulti2))
       #DISPLAY ('Procedure Name')
       #PROMPT  ('',@s255),%ProcedureNameMulti,Default('Browse'),REQ,At(,,180,)
    #ENDBUTTON
#ENDTAB
#ENDSHEET
#!---------------------------------------------------------
#! Add the procedures to the map
#!---------------------------------------------------------
#AT(%GlobalMap)
#IF(%DisableAll=%False)
#SET(%UModule,UPPER(%Module))
MODULE('%UModule')
  #INDENT(-5)
  #FOR(%LookupFileMultiCount)
    #SET(%ValueConstruct,%ProcedureNameMulti)
%[32]ValueConstruct PROCEDURE
  #ENDFOR
  #INDENT(5)
END
#ENDIF
#ENDAT
#!-----------------------------------------------------------
#! Add the procedures to the export list
#!-----------------------------------------------------------
#AT(%DLLExportList)
#IF(%DisableAll=%False)
  #FOR(%LookupFileMultiCount)
    #SET(%ValueConstruct,%ProcedureNameMulti & '@F')
  %[55]ValueConstruct @?
  #ENDFOR
#ENDIF
#ENDAT
#!---------------------------------------------------------
#! Generate the Browse:LookupFile procedures
#!---------------------------------------------------------
#AT(%ProcessedCode),PRIORITY(7500)
#IF(%DisableAll=%False)
#FOR(%LookupFileMultiCount)
#!---------------------------------------------------------
%ProcedureNameMulti    PROCEDURE
!----------------------------------------------------------
szCaption   CString(256)
szProcName  CString(256)
View:Browse View(%LookupFileMulti)
               Project(%LookupFieldIDMulti1)
               Project(%LookupFieldIDMulti2)
            END
  CODE
  szCaption   = '%ScreenCaptionMulti'
  szProcName  = '%ProcedureNameMulti'
  %SourceLookupProcName(szCaption,  View:Browse, Relate:%LookupFileMulti, |
    %LookupNumericKeyMulti, %LookupAlphabeticKeyMulti, |
    %LookupFieldIDMulti1, %LookupFieldIDMulti2, szProcName)
  RETURN
#ENDFOR
#ENDIF
#ENDAT
#!--------------------------------------------------------
#GROUP(%SETCC)
#!---------------------------------------------------------
#BOXED('DinamiComp Template Set'),SECTION,AT(,10,,65)
  #DISPLAY ('')
  #DISPLAY ('DinamiComp CC Lookup Template Set for Clarion 6.3')
  #DISPLAY ('Copyright 2006 All Rights Reserved WorldWide.')
  #DISPLAY ('')
  #DISPLAY ('Version 1.0')
#ENDBOXED
#!---------------------------------------------------------
#GROUP(%SETAPP)
#!---------------------------------------------------------
#BOXED('DinamiComp Lookup Templates'),SECTION,AT(,10,,45)
  #DISPLAY ('')
  #DISPLAY ('Automatic Lookup-File Procedure Creation')
  #DISPLAY ('')
#ENDBOXED
#!---------------------------------------------------------
#GROUP (%StripExclamation,*%zField)
#!---------------------------------------------------------
  #IF(SUB(%zField,1,1)='!')
    #SET(%zField,SUB(%zField,2,LEN(%zField)-1))
  #ENDIF
```

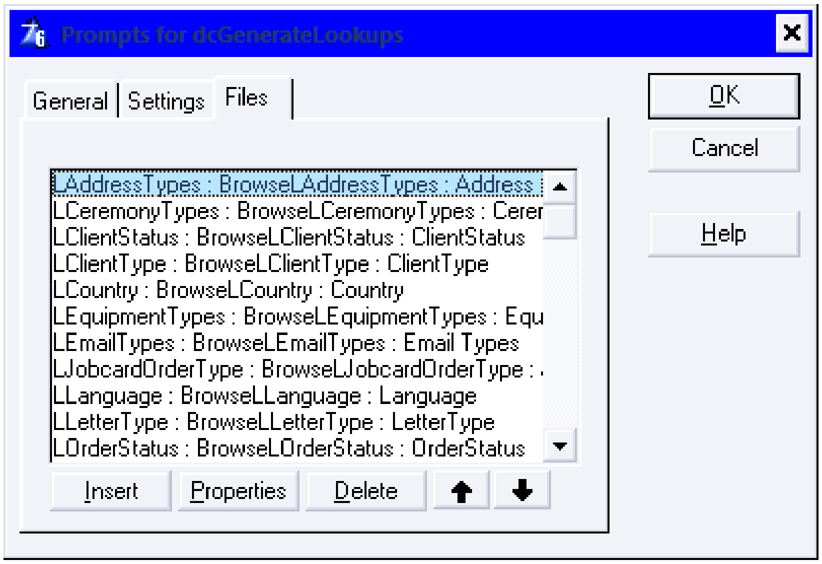
I registered my new template and attached it to the `GenerateLookupBrowse` procedure.



Tables
Data
Embeds
%DataSection:
SOURCE (INCLUDE('ABBROWSE.INC'),ONCE)
%ProcessedCode:
SOURCE (GlobalResponse = ThisWindow.Run()
Extension
DinamiComp's Generate Lookup Procedures-Procedure

Figure 1. Source Procedure

When you double click on DinamiComp's Generate Lookup Procedures Template you are presented with the screen in Figure 2.

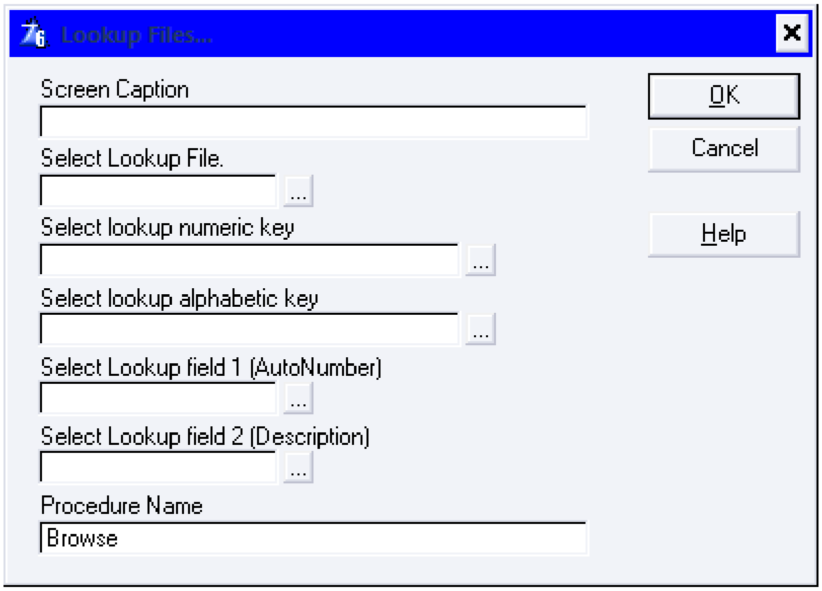


LAddressTypes : BrowseLAddressTypes : Address
LCeremonyTypes : BrowseLCeremonyTypes : Cerer
LClientStatus : BrowseLClientStatus : ClientStatus
LClientType : BrowseLClientType : ClientType
LCountry : BrowseLCountry : Country
LEquipmentTypes : BrowseLEquipmentTypes : Equ
LEmailTypes : BrowseLEmailTypes : Email Types
LJobcardOrderType : BrowseLJobcardOrderType :
LLanguage : BrowseLLanguage : Language
LLetterType : BrowseLLetterType : LetterType
LOrderStatus : BrowseLOrderStatus : OrderStatus

Figure 2. Template Prompts – Files Tab

On the Files tab you populate the tables for which you want common browse screens. The information that you need to for each record, as shown in Figure 3, is as follows:

- Screen Caption – the screen caption of the browse screen for this file
- Lookup File – the file that want to browse and update
- Numeric key – the key for the autonumber field
- Alphabetic key – the key for the description field
- Lookup Field 1 – the autonumber field
- Lookup Field 2 – the description field
- Procedure Name – the name of the procedure that will be generated. You will use this name to call this procedure from the main menu.



Screen Caption

Select Lookup File.

Select lookup numeric key

Select lookup alphabetic key

Select Lookup field 1 (AutoNumber)

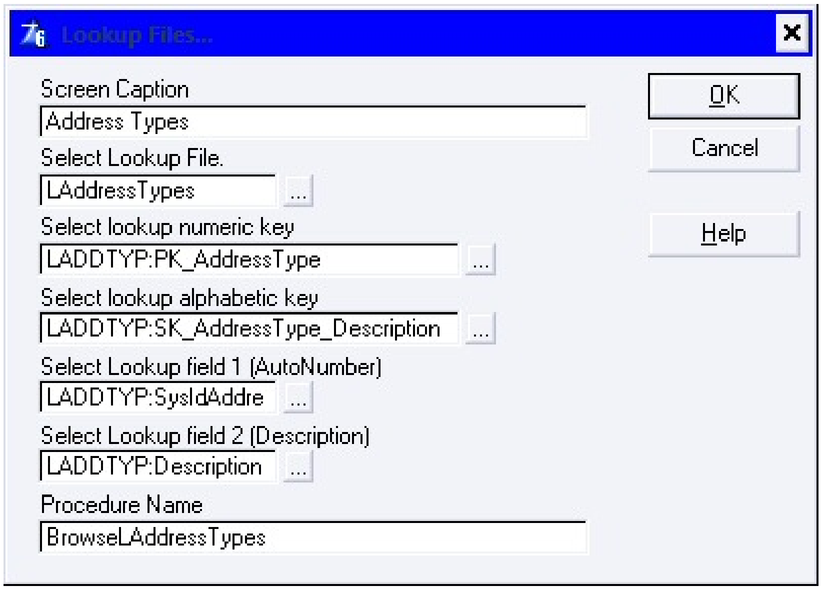
Select Lookup field 2 (Description)

Procedure Name
Browse

Figure 3. Template Prompts – Individual Items

I know this is a lot of information but it gives you full control over what happens with the common browse.

Figure 4 shows what a populated screen looks like:



Screen Caption
Address Types

Select Lookup File.
LAddressTypes

Select lookup numeric key
LADDTYP:PK_AddressType

Select lookup alphabetic key
LADDTYP:SK_AddressType_Description

Select Lookup field 1 (AutoNumber)
LADDTYP:SysIdAddre

Select Lookup field 2 (Description)
LADDTYP:Description

Procedure Name
BrowseLAddressTypes

Figure 4. Template Prompts – Populated Screen

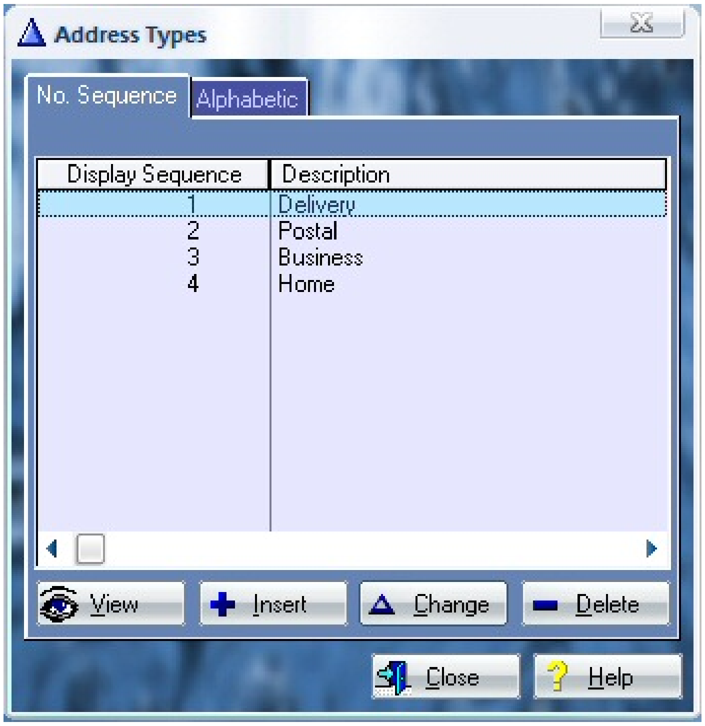
The template generates a procedure for each entry in its list. Here is the code for one such procedure:

```
BrowseLAddressTypes    PROCEDURE
szCaption   CString(256)
szProcName  CString(256)
View:Browse View(LAddressTypes)
               Project(LADDTYP:SysIdAddressType)
               Project(LADDTYP:DescriptionAddressType)
            END
  CODE
  szCaption   = 'Address Types'
  szProcName  = 'BrowseLAddressTypes'
```

```
    GenerateLookupBrowse(szCaption,  View:Browse, Relate:LAddressTypes, |
      LADDTYP:PK_AddressType, LADDTYP:SK_AddressType_Description, |
      LADDTYP:SysIdAddressType, LADDTYP:DescriptionAddressType, szProcName)
    RETURN
```

Finally, I added menu options to my application frame in the EXE application to call each of the generated browses, and then I compiled the application.

This is what the above browse looks like in my application (look and feel by Capesoft MakeOver):



Display Sequence	Description
1	Delivery
2	Postal
3	Business
4	Home

Figure 5. Sample Browse

What do you need to do to add more lookup tables?

1. Open your dictionary and add the new table to your dictionary.
2. Open the Data application, go to the `GenerateLookupBrowse` procedure, open DinamiComp’s Generate Lookup Procedures Template and add the new Lookup table info. Remember to make a note of the procedure name.
3. Compile the Data DLL.
4. Open the Main application, open the frame procedure and add the new Lookup table as a menu item with the procedure name as in step 4.
5. Double click on the ToDo procedure and specify it as external.
6. Compile your main EXE, run the application and test the new lookup browse

Features to be added

I could not get the features on the browse templates’ Extended Options tab (including sort headers) to work. Perhaps someone would like to take that on.

Summary

With a little work up front, you can easily add new lookup tables to your application. If you want to modify the look and feel of the lookup browses, you only need to change the one browse procedure and it is done. Randy has put it so nicely in his article: "The entire application is considerably smaller because of this approach, and you have relieved yourselves of the repetitious and boring task of fine tuning wizard-generated browses."

Download the source

Nardus Swanevelder was born and raised in South Africa. He was a networking engineer for seven years before he moved over to the commercial side of the business. Nardus has developed a Sale Cycle Management system for the Information and Communication Technology industry. He has been programming in Clarion since 1989, and holds B.Com and MBA degrees. In his spare time Nardus lectures Financial Management to B. Com Hons students at North-West University.

Reader Comments

Add a comment

Clarion Magazine

A Class Wrapper for the SimpleOCR API

by John Dunn

Published 2006-07-06

Recently I completed a document management system that required the indexing of key information from the documents scanned into the system. In order to reduce data entry time and increase the accuracy of the indexed data I decided to add Optical Character Recognition (OCR) capabilities to the program.

Since there were no readily available OCR tools for Clarion, I decided to look for a reasonably priced OCR library that would meet the needs of my program. After some searching I found two libraries that looked promising. The first library that I tested was the OCR Developer's Toolkit from CardReader Inc. I tested version 8.0 but found quite a few inaccuracies in the documentation, which they readily admitted had not been fully updated from version 7.0. More discouraging, however, were the numerous errors I received attempting to scan various .tif images that had more than one bit per pixel resolution.

SimpleOCR

The second OCR library I tested was the SimpleOCR API from Simple Software. They advertise their library as the only royalty free OCR API. This was very appealing to me. My evaluation of this library was very positive so I decided to purchase the unlimited license. The SimpleOCR API is offered as a DLL or as an ActiveX control called SimpleOCX. I decided to use the DLL interface since it is a more direct link to the functions provided in the library.

The SimpleOCR API provides a large number of functions that the documentation categorizes into the following groups:

- Tiff File Manipulation
- Image Management
- OCR Functions
- Scanner Functions
- Set of Images Management

Since I am using ImageEx for scanning, image manipulation and management, I mainly concentrated on implementing the OCR functions in the SimpleOCR library. That being said, I still needed to implement a few of the file and image management functions that are required by the OCR process, such as the Load Image

(`LoadImg`) and Load Multiple Images (`LoadMultipleImg`) functions.

Class wrapper

My aim was to write a class wrapper to encapsulate the various OCR functions and data. Before doing that, I created a small procedural program to test each of the functions I needed one at a time. My first task was to use LibMaker to create a Clarion compatible .lib file from the ocrdll.dll that is supplied as a part of the SimpleOCR API package. I assembled all of the SimpleOCR functions that I needed in my program and converted the prototypes to Clarion. Here is the OCRDLL.Lib module I created:

```
                        MODULE('OCRDLL.Lib')
    AddImage              PROCEDURE(LONG plImageSet,LONG plImage),SHORT,|
                                    RAW,PASCAL
    CreateMultipleImg     PROCEDURE()LONG,RAW,PASCAL
    DeskewImg             PROCEDURE(LONG plImage),LONG,RAW,PASCAL
    DIBToImg              PROCEDURE(LONG plhDIB),LONG,RAW,PASCAL
    EraseBlackBordersImg  PROCEDURE(LONG plImage),LONG,RAW,PASCAL
    ExtractImgArea        PROCEDURE(LONG plImage,LONG plXPos,LONG plYPos,|
                                    LONG plWidth,LONG plHeight),LONG,RAW,PASCAL
    FreeImg               PROCEDURE(LONG plImage),RAW,PASCAL
    FreeMultipleImg       PROCEDURE(LONG plImageSet),RAW,PASCAL
    GetNbImages           PROCEDURE(LONG plImageSet),LONG,RAW,PASCAL
    GetImage              PROCEDURE(LONG plImageSet,LONG plIdx),LONG,|
                                    RAW,PASCAL
    GetImgRes             PROCEDURE(LONG plImage,*LONG plHRes,*LONG plVRes),|
                                    RAW,PASCAL
    GetImgSize            PROCEDURE(LONG plImage,*LONG plWidth,*LONG plHeight),|
                                    RAW,PASCAL
    LoadImg               PROCEDURE(LONG plFilenameAddress),LONG,RAW,PASCAL
    LoadMultipleImg       PROCEDURE(LONG plFilenameAddress),LONG,RAW,PASCAL
    OCR                   PROCEDURE(LONG plImage,LONG plNoisy),LONG,RAW,PASCAL
    OCRSetProgressHandler PROCEDURE(LONG plProc),LONG,RAW,PASCAL
    OCRSetOutputHandler   PROCEDURE(LONG plProc),LONG,RAW,PASCAL
    RotateImg             PROCEDURE(LONG plImage,LONG plAngle),LONG,RAW,PASCAL
    SaveImg               PROCEDURE(LONG plFilenameAddress, LONG plImage),LONG,|
                                    RAW,PASCAL
    SaveMultipleImg       PROCEDURE(LONG plFilenameAddress, LONG plImage),LONG,|
                                    RAW,PASCAL
    SetLanguage           PROCEDURE(LONG plLanguage,LONG plDictDir),RAW,PASCAL
    SetOutputMode         PROCEDURE(LONG plMode),RAW,PASCAL
    SetOutputWindow       PROCEDURE(LONG plhWnd),RAW,PASCAL
                        END ! module
```

In addition to the above OCRDLL.DLL interface module I had to prototype and create output and progress handler callback procedures. The output handler allows the program/class to manage output events and data that has been processed by the OCR engine. There are a number of events that are generated by the OCR engine but I only needed three of them:

- The `eOT_Text` event indicates that a character has been recognized and its ASCII value is

available to be processed. This is where I add processed characters to the output string.
- The eOT_ENDL event indicates that an end of line has been reached. When this event is received I add CR/LF characters to the output string.
- The eOT_ENDZ event indicates that the end of a text area has been reached. As with the end of line event, I also add CR/LF characters to the output string when this event is received.

The progress handler allows the OCR engine to display the progress of the OCR job. If a progress control exists on a screen that interfaces with the OCR class, the field equate value of the progress control is passed to the class and used by the progress handler.

Callbacks

The SimpleOCR output and progress handler functions are armed by passing the address of the output and progress handler procedures to the library's OCRSetOutputHandler and OCRSetProgressHandler functions:

```
SELF.lOutputHandlerAddress = ADDRESS(OutputHandler)
SELF.lOutputHandler = |
  OCRSetOutputHandler(SELF.lOutputHandlerAddress)
SELF.Trace(eMethodName & 'lOutputHandlerAddress = ' |
  & SELF.lOutputHandlerAddress)
SELF.Trace(eMethodName & 'lOutputHandler = '        |
  & SELF.lOutputHandler)
IF SELF.lProgressFEq ~= 0
  SELF.lProgressHandlerAddress = ADDRESS(ProgressHandler)
  SELF.lProgressHandler = |
    OCRSetProgressHandler(SELF.lProgressHandlerAddress)
  SELF.Trace(eMethodName & 'lProgressHandlerAddress = ' |
    & SELF.lProgressHandlerAddress)
  SELF.Trace(eMethodName & 'lProgressHandler = '        |
    & SELF.lProgressHandler)
END ! if
```

The SimpleOCR API can be set to process the following languages out of the box:

- American English
- UK English
- French
- Dutch
- Custom

Four modifiable dictionary files are provided that have the extension of .wdc. Two support programs allow you to list the contents of a dictionary file and to add words to the dictionary: WDC2List.Exe. and List2WDC.Exe. The accuracy of the OCR engine can be enhanced by the use of the appropriate dictionary file. There is also an option to use no dictionary at all if so desired. The SetLanguage function allows you to tell the OCR engine which dictionary file to use. Before calling this function the library needs to know the directory in which the dictionary file can be found.

Character recognition

The OCR library allows you to load single and multiple page .tif (Tagged Image File Format) files. The images to be processed can be cleaned up by several useful functions that will enhance the accuracy of the conversion. The deskew function will attempt to 'straighten out' the graphical elements in an image so that they can be better interpreted by the OCR engine. Another function will attempt to 'reduce noise' or the small specs and marks that are sometimes picked up in the scanning process. When converting text on forms, the 'erase black borders' function can be helpful in increasing OCR accuracy. There is no hard and fast rule about which of these methods will help improve the chances of getting a clean conversion to text. Generally, the cleaner and crisper the scanned image is the better the OCR results will be. In the attached example program I provide a way to configure these image manipulation options at run time.

The example program demonstrates the use of my OCR wrapper class. You will need to download the API demo from SimpleOCR and install it. This will provide the ocrdll.dll, dlltwain.dll and dictionary (.wdc) files you will need to run the example program. The demo will also provide several .tif image files that can be used to test the OCR functionality.

The implementation of the class in an application or program is very simple. The class definition (OCRClass.Def) and include file (OCRClass.Inc) must be included in the global section of the program. In an app I generally place these include statements in the 'After Global Includes' embed.

```
INCLUDE('OCRClass.Def'),ONCE
INCLUDE('OCRClass.Inc'),ONCE
```

Two project defines need to be added to the project global options defines tab:

```
_OCRDllMode_=>0
_OCRLinkMode_=>1
```

If these are not added to the project your program will generate an immediate GPF. The settings above indicate that the class will be linked into the program being compiled.

The OCR object can be instantiated in many ways but I generally declare it in the data section like this:

```
oOCR cOCR
```

Before calling the main OCR method to process an image, the language, dictionary and various OCR properties should be set. If you are displaying the processed text and/or a progress bar the field equates of these controls need to be set also. In the example program this is done just after the `CODE` statement, as follows:

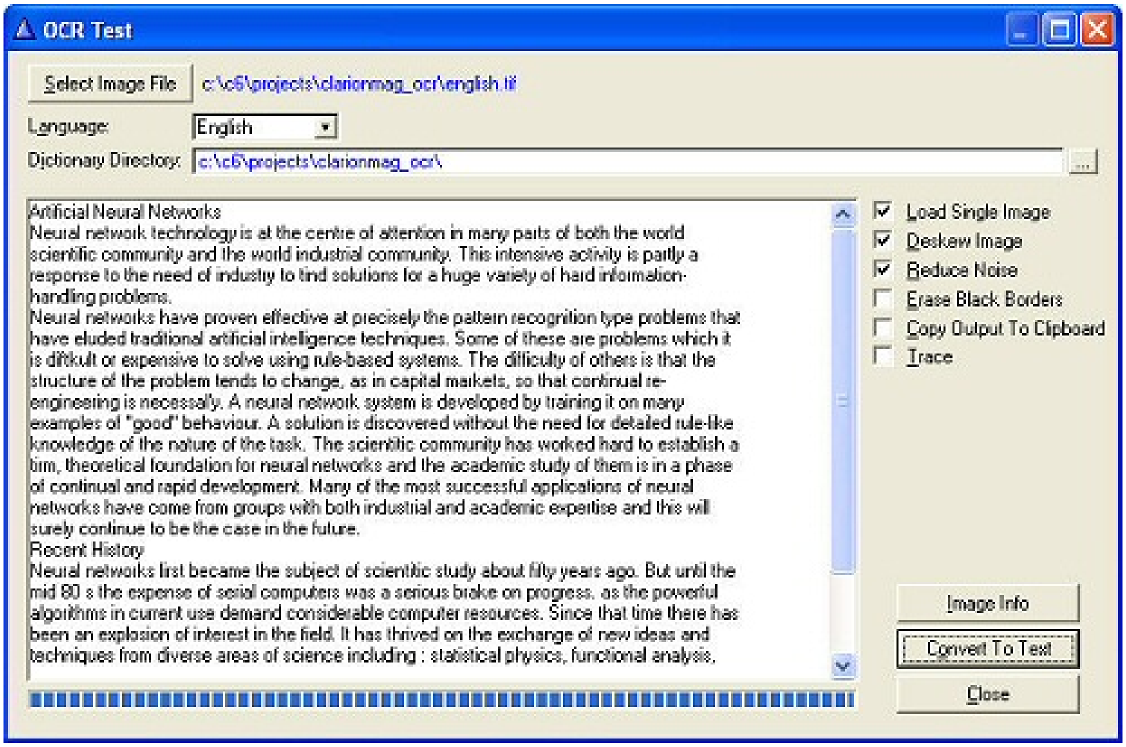
```
bLanguage                = GETINI(eINISection,'Language',|
                             eEnglish,eINIFileName)
sDictDir                 = GETINI(eINISection,'DictDir',|
                             oOCR.ProgramDir(True),eINIFileName)
sInputFile               = GETINI(eINISection,'InputFile',|
                             '',eINIFileName)
oOCR.bOutputToClipboard = GETINI(eINISection,'OutputToClipboard',|
                                    '0',eINIFileName)
    oOCR.bLoadSingleImage   = GETINI(eINISection,'LoadSingleImage',|
                                    '1',eINIFileName)
    oOCR.bReduceNoise       = GETINI(eINISection,'ReduceNoise',|
                                    '1',eINIFileName)
    oOCR.bDeskewImage       = GETINI(eINISection,'DeskewImage',|
                                    '1',eINIFileName)
    oOCR.bEraseBlackBorders = GETINI(eINISection,'EraseBlackBorders',|
                                    '0',eINIFileName)
    oOCR.bTraceOn           = GETINI(eINISection,'TraceOn',|
                                    '0',eINIFileName)
    oOCR.bLanguage          = bLanguage
    oOCR.sDictDir           = sDictDir
    oOCR.lProgressFEq       = ?bProgress
    oOCR.lOutputTextFEq     = ?sOutputText
```

The main OCR processing is done in the OCR method. You pass the image file path/name and the page number to this method. The page number is needed for processing multipage .tif images. You will notice that I have made some provisions for processing Device Independent Bitmaps (DIB) that may have been loaded into memory. This functionality is not complete at this point in time, so the second parameter to pass the handle of the DIB is simply set to zero.

You will also notice that there is an option to turn on debug tracing. When this is option is checked, debugging information will be output to the system debug log. You can use DebugView from SysInternals to view the trace/debug messages as they are being processed.

In my working copy of the class, I implemented a few methods from a couple of other classes. I have removed those dependencies and included some of those methods directly in this class. The constructor and destructor methods were used to instantiate and dispose of these implemented classes but since they are not a part of this version of the OCR class they do not really do anything. I decided to leave them for tracing purposes as there are trace calls which will make it clear in the debug log when the object is instantiated and destroyed.

Figure 1 shows a screen shot of the example program that includes all of the major elements used in the OCR process, and which shows the results of the OCR conversion after it has processed the ‘English.Tif’ image file:



Select Image File: c:\c6\projects\clarionmag_ocr\english.tif

Language: English

Dictionary Directory: c:\c6\projects\clarionmag_ocr\

☑ Load Single Image
☑ Deskew Image
☑ Reduce Noise
☐ Erase Black Borders
☐ Copy Output To Clipboard
☐ Trace

```
Artificial Neural Networks
Neural network technology is at the centre of attention in many parts of both the world
scientific community and the world industrial community. This intensive activity is partly a
response to the need of industry to find solutions for a huge variety of hard information-
handling problems.
Neural networks have proven effective at precisely the pattern recognition type problems that
have eluded traditional artificial intelligence techniques. Some of these are problems which it
is diffkult or expensive to solve using rule-based systems. The difficulty of others is that the
structure of the problem tends to change, as in capital markets, so that continual re-
engineering is necessaly. A neural network system is developed by training it on many
examples of "good" behaviour. A solution is discovered without the need for detailed rule-like
knowledge of the nature of the task. The scientific community has worked hard to establish a
tim, theoretical foundation for neural networks and the academic study of them is in a phase
of continual and rapid development. Many of the most successful applications of neural
networks have come from groups with both industrial and academic expertise and this will
surely continue to be the case in the future.
Recent History
Neural networks first became the subject of scientific study about fifty years ago. But until the
mid 80 s the expense of serial computers was a serious brake on progress. as the powerful
algorithms in current use demand considerable computer resources. Since that time there has
been an explosion of interest in the field. It has thrived on the exchange of new ideas and
techniques from diverse areas of science including : statistical physics, functional analysis,
```

Figure 1. OCR test program

The Load Single Image option, when checked, will only load page one of a multipage .tif file. When this is unchecked all pages of a multipage .tif file will be loaded and processed. In my document management program I allow the user to specify the page they want to convert to text. The Image info button displays the resolution and size of the selected image.

My experience of the SimpleOCR engine is that it is very solid and reliable. Clearly scanned images of 200 to 300 DPI convert quite well without the need to deskew or reduce noise. The demo package includes a skewed image that converts nicely with the deskew option in effect.

The SimpleOCR API is well documented and quite easy to implement in Clarion. My class wrapper makes it even easier to pop OCR into an application. The unlimited license is not inexpensive ($2850 as of this writing), but it is royalty free. You can purchase an organizational license for $1150, which lets you distribute the API to all users within your company, or for full control you can get a source code license for $5000.

Download the source (updated July 22)

John Dunn is an independent software developer who has been programming in Clarion since the first DOS version. He and his wife live in Bolton, Massachusetts. They have two daughters who are attending college. When not at working at his computer, John enjoys reading, cooking, exercising, listening to live music and viewing films.

Reader Comments

Add a comment

- » Dave included the wrong source file with the article. ...
- » Sorry about that everyone - I've updated the...

Clarion Magazine

The ClarionMag Blog

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

XML All blog entries
XML All new items, including blogs

Blog Categories

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

June PDF

Direct link

Posted Monday, July 31, 2006 by Dave Harms

It's been brought to my attention that I neglected to get the June PDF up on the site before I left on an extended road trip. Unfortnately I didn't bring the necessary version of Acrobat with me, so the PDF will have to wait until I'm back next week. Thanks for your patience.

C7/Clarion.NET UK CUG commentary - UPDATED

Direct link

Posted Monday, July 31, 2006 by Dave Harms

I'm not at the UK CUG meeting, so I don't have first hand information on what's been presented, but going by DevDawn's report there are a few issues that I don't think have been reported before. These include:

- IDE throwing an exception on an error, rather than a GPF. This lets you continue working, which is nice, but if you have an IDE activity that throws an exception, you still have a problem.
- I'm not quite sure what's meant by the comment on the Win32 compiler. Evidently the Clarion Win32 compiler has been rewritten as a .NET application, which makes sense given the IDE is .NET.
- Multiple users in the IDE is a nice idea, though I'm not sure it will be used much. How many people share the IDEs?
- Dunno what the source control comment means - version control?
- Inplace activation of ActiveX components has been mentioned before, but it's nice to see it confirmed.
- Support for .NET Compact Framework is good news.
- Having to open multiple copies of the IDE to have multiple apps open is a bit of a disappointment.
- It appears that although you can use TPS files in Clarion.NET, there is (at least so far) no ADO.NET provider for TPS files, meaning that you will not be able to get access to TPS files from other .NET languages. This seems a bit bizarre to me, since if you can create a Clarion.NET assembly that access TPS files, you must surely be able to use that assembly from any other .NET application. But as long as there's no ADO.NET provider for TPS files, any such access from other apps will be a bit clunky. UPDATE: Mark Sarson has kindly clarified that the TPS .NET driver will only be available with Clarion.NET and not as a separate product.
- Partial class support in Clarion.NET. Partial classes are classes that can be split across multiple physical files. It's another organization technique that lets different developers (or different tools) access one part of a class without munging up the other part. It's also a potential alternative or companion to the current embed system...

- Clarion.NET source files have the extension .cln
- Clarion.NET includes a C# to Clarion.NET converter.
- .NET 3.0 support. Well, this really doesn't mean that much, since .NET 3.0 is really .NET 2.0 with some extra class libraries. So it's not like the core stuff has changed in a big way.
- LINQ - hmm, I wonder what they're planning here. LINQ is some cool query technology that lets you work with databases, XML files, lists etc using a common syntax.
- 64 bit native comilers - well, this really should be one of the benefits of going to .NET. You can piggyback on existing technology.
- People don't want they hand-coder's .NET edition? Hoo boy.

More OSCON to come

Direct link

Posted Friday, July 28, 2006 by Dave Harms

OSCON has wrapped up, and I'm working on getting a number of reports up on the site. Some of these will be free access, others will be subscriber-only.

I was quite impressed by the conference organization and the quality of speakers. There was a fair bit of meat there even for someone like me coming from the Clarion world (thought with an admitted bias toward certain open source tools such as Java.) I mainly followed the database track, which for the most part meant PostgreSQL and MySQL. I can't say I even heard Firebird mentioned, which surprised me a little.

The keynote addresses were generally informative, frequently entertaining, and in one case tear-wipingly funny.

More coverage on the way! But as I'm going to be back on the road next week, I probably won't get the last items posted until the second week of August.

Winternals/Sysinternals bought by MS

Direct link

Posted Thursday, July 27, 2006 by Dave Harms

Winternals and SysInternals have been bought by Microsoft. For now, the sites will stay as they are. Translation: download the tools now!

Google Announces Open Source Hosting Service

Direct link

Posted Thursday, July 27, 2006 by Dave Harms

Google's big OSCON announcement is, as expected, a simplified, AJAX-based project hosting service for open source projects

Google has rebuilt Subversion on BigTable, which is a "massively scalable failover blah blah blah (technical term - ed.) storage system."

The Google team also did a rethink of issue tracking. There is a lot of workflow structure in current issue tracking systems. Much of this has been replaced by labels, which can be tracked using the Google search code. More to follow - session is still underway.

A Google open source repository?

Direct link

Posted Thursday, July 27, 2006 by Dave Harms

Rumors are flying that this afternoon at OSCON Google will announce an open source repository like SourceForge. Stay tuned.