

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

June 2000 Index

[Main Page](#)[COL Archive](#)[Log In](#)[Subscribe](#)[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)[Article Index](#)[Author Index](#)[Links To](#)[Other Sites](#)[Downloads](#)[Open Source](#)[Project](#)[Issues in](#)[PDF Format](#)[Free Software](#)[Advertising](#)[Contact Us](#)

[ETC Wrapup: Sights And Sounds](#)

A few more notes and pictures from ETC 2000.
(Jun 6,2000)

[ETC Presentation Summary: James Fortune](#)

James Fortune started the conference off with a completely new version of his famous help presentation.
(Jun 6,2000)

[ETC Presentation Summaries: Bruce Johnson, Andy Stapleton](#)

Summaries of two all-day presentations: Andy Stapleton on taking Sybase SQL to the web, and Bruce Johnson's class on understanding ABC.
(Jun 6,2000)

[COL Updates To Resume Shortly](#)

We've been working on re-releasing COL articles at a rate of one issue per week. At present, however, we're making some improvements to the software used to produce and deliver Clarion Magazine. Among other things, these improvements will make it easier for us to catalog and publish the two years worth of material in the COL archive. We'll resume the COL articles as soon as possible. In the meantime, if you have a specific article you want to see please email editor@clarionmag.com and we'll put it on the priority list.
(Jun 6,2000)

[ITEMIZE Your Parameters For A Better OMITTED\(\)](#)

Omitted parameters can be handy, but they're also an easy way to add bugs to your program. Carl Barnes outlines a strategy for bug-free omits.
(Jun 14,2000)

[The Clarion Advisor: Standard Address Handling](#)

Steve Parker looks at strategies for handling addresses with variable numbers of lines.
(Jun 14,2000)

[Clarion For Zebras: Working With Barcodes \(Part 1\)](#)

Simon Brewer explains the ins and outs of using barcoding technology with Clarion. Part 1 of 2.
(Jun 20,2000)

[True Confessions: A Tale of Two Users](#)

The cost of knowing what you are doing is having learned the hard way. Somehow, it doesn't matter how many books, manuals, articles or true confessions you read, you never really learn until you've made the painful mistake yourself.

(Jun 20,2000)

[The Clarion Advisor: EIP Tips](#)

Have you ever needed your edit-in-place fields to enforce upper case or capitalization? Here's how.

(Jun 20,2000)

[MySQL/MyODBC Update](#)

Here's the latest on MySQL including MyODBC driver issues, plus a few corrections of fact.

(Jun 27,2000)

[Clarion For Zebras: Working With Barcodes \(Part 2\)](#)

Simon Brewer explains the ins and outs of using barcoding technology with Clarion. Part 2 of 2.

(Jun 27,2000)

[Tool Talk: Getting Into SQL On The Cheap](#)

Think SQL is expensive? Not necessarily. Tom Hebenstreit shows where to get free or inexpensive developer versions of popular SQL database servers.

(Jun 27,2000)

[June 2000 News](#)

Clarion news, notes, and happenings from around the globe.

(Jun 27,2000)

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

ETC Wrapup

by Dave Harms

The second East Tennessee Clarion conference was, if it's possible, even more memorable than the first. I've been privileged to attend quite a few regional Clarion conferences, and recently these small gatherings have risen well above the standard that might be expected of them. To my mind, it was Kathryn and Phil Carroll who started the trend with their landmark Arizona DevCon. Kathryn showed that it was possible have a regional conference with real style (and really good food!). Lee White then turned an informal gathering of developers into the first ETC in 1998, at a riverside hotel in the picturesque town of Gatlinburg, in the Smoky Mountains.

Later that year I had an opportunity to attend a regional Clarion conference (ConVic '98) at an estate outside of Melbourne, Victoria, Australia. Apparently the small scale, high quality Clarion conference isn't a purely North American phenomenon, as Chris Livingstone & co. demonstrated. And now this year Lee White reprised ETC, a little bigger and a even a little better than the first time.

So what is it about these mini-DevCons that makes them so good? I've already mentioned food, and there's no doubt that the Cajun Cookouts hosted by Andrew and Sabrina Guidroz in Phoenix and Gatlinburg (and elsewhere!) have been significant factor. You wouldn't believe the gear they haul around in that pickup truck, or the resulting feasts. (They weren't at ConVic, but then Chris had arranged for a truly exceptional level of catering, so good vittles weren't exactly in short supply, nor was there a dearth of Australian beer and wine.)

Food, location, the odd wild animal or near-natural disaster, all these things add to the mix. Sometimes the conference format is different – at ETC, for instance, there is only one presentation track, and each session is three hours in length, which makes for more in-depth treatment. Ultimately, though, it's the people.

Go to any conference and you'll see a lot of the usual suspects. There are well-known TTS members, third party developers, the occasional Clarion investor and/or board member, TS/SV people (usually), newsgroup agitators, chow hounds and beer samplers, and Friends of Clarion. And thankfully, there are always new faces too. If you're a new

[ETC Wrapup: Sights And Sounds](#)

(Jun 6, 2000)

[ETC Presentation Summary: James Fortune](#)

(Jun 6, 2000)

[ETC Presentation Summaries: Bruce Johnson, Andy Stapleton](#)

(Jun 6, 2000)

[June 2000 News](#)

(Jun 6, 2000)

[COL Updates To Resume Shortly](#)

(Jun 6, 2000)

Developer
PLUS

Your source for
development tools
and add-ons.

Your outlet for
application sales.

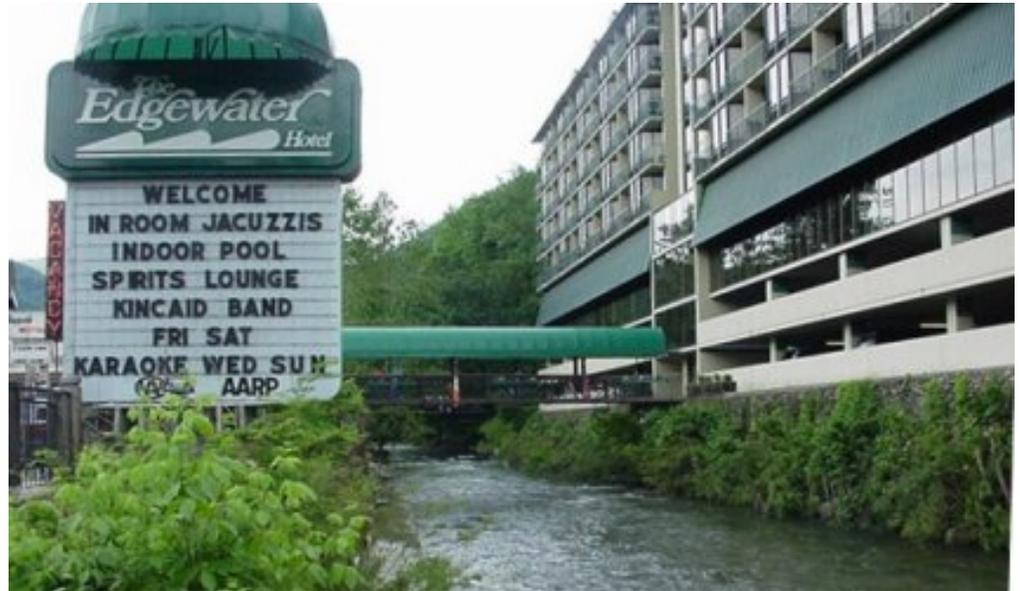
face, keep coming back. Let folks watch you get older.

Maybe it's because Clarion has never been a corporate mainstay. Many of us work in small shops or on our own, and most of our contact with other Clarion developers is by phone, email, ICQ, or the newsgroups. DevCons are where we put faces to the names, renew old friendships, and tell stories. And eat crawfish etouffee.

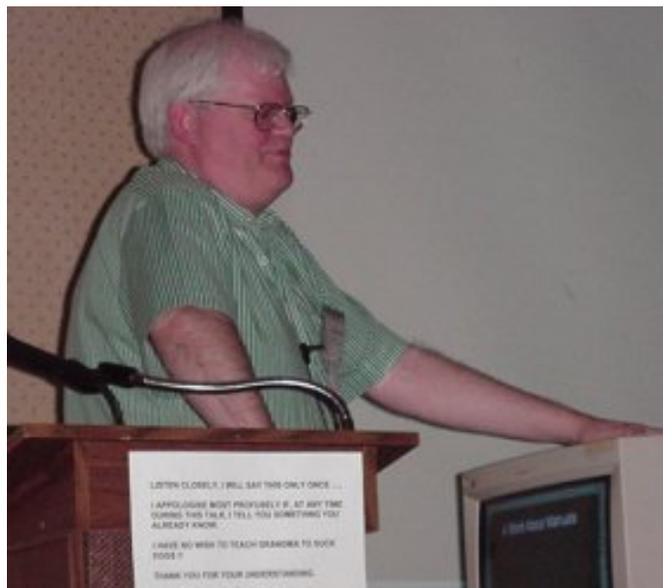
To the many people who put on DevCons, and this time around to Bob Dobbins and the Knoxville Clarion User's Group, and especially to Lee White, thank you.

A Little Something Extra

Here are a few more pictures from ETC2000. And for those of you with the amazing tilt-a-whirl monitors, some more sideways MPEGs!

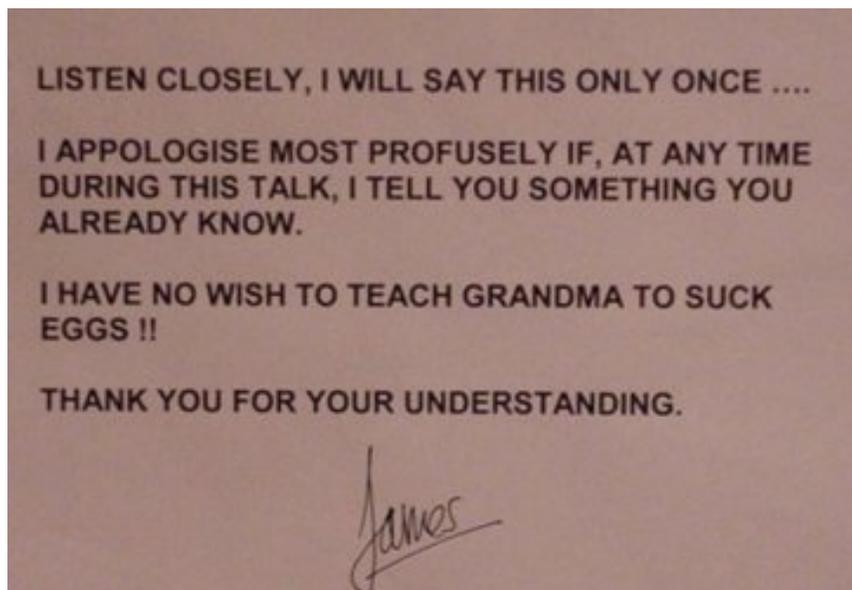


The Edgewater hotel on Thursday, after the water had started to recede again. Sessions were held at the far end of the hotel.



James Fortune in his pre-conference session.

Podium note shown below...



Those hoping for a jug band to kick off the festivities Tuesday night were sadly disappointed.



Bob Zaukere, SoftVelocity's President and CEO, speaks to an entirely awake room.



"I'm telling you, we did it. Black bear etouffee is now on the menu."

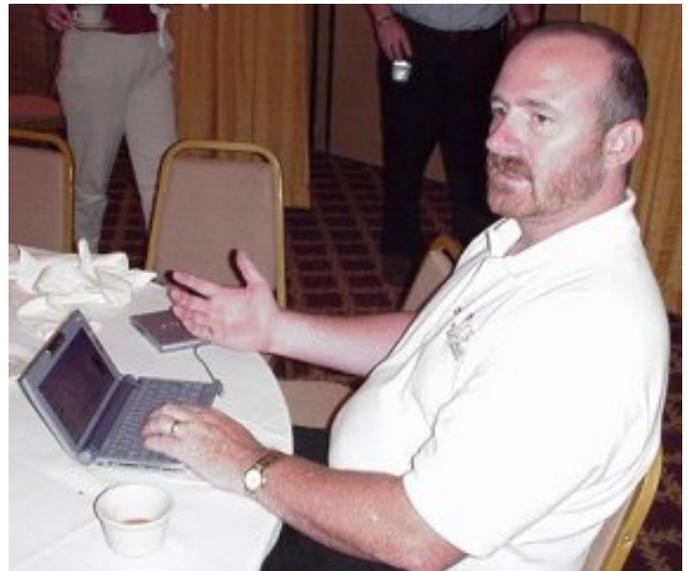


If you look closely at the screen behind Skip, you'll notice he's about to demonstrate some secret new technology...



"I ate black bear what?"

Cowboy really is a larger-than-life character. That's a *normal-sized laptop* he's using.





Dr. Parker gives his steamboat gambling lecture.

Some Doubtful MPEGs

And, of course, the remaindered MPEGs. WARNING: These MPEGs either suffer from poor video, inferior audio, or were shot sideways (which is close to what will happen to the videographer next time he does this). View the axially challenged MPEGs only after consulting with your chiropractor. No slur is, of course, intended toward the subjects of these videos.

Why are we putting these up? Because some of you are video hounds with more bandwidth than is probably good for you. Enjoy.

- Skip Williams talks about a Clarion FTP client ([864K](#))
- Bob Zauhere on the relationship between SV and Sensium - sideways ([784K](#))
- Bob on Data Junction - sideways ([1168K](#))
- Bob on ways to reduce the number of embed points shown to the user - sideways ([480K](#))
- James Fortune explaining how not to set up install programs ([544K](#))
- Raffleman 2000 losers ([792K](#))
- Nik Johnson on dictionary user options and handling CSTRINGs - sideways ([1127K](#))

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)

ETC Presentation Summaries

by Dave Harms

James Fortune: User Assistance 2000

James Fortune kicked off ETC with an early bird session devoted not so much to writing help, as to avoiding writing help. As James pointed out, we don't like to write help files, and users don't like reading them. And do users often need help? What about web sites, which almost never have any associated help?

The point, of course, is that the better an application's design, the less likely the user is to need help. And mostly that means coding the user interface according to standards, starting right from the user's first interaction with the program. Install programs should always work properly with the default options. Your program can also have useful default behavior, such as opening a most-used browse when the program loads.

Good Windows application design means adhering to Microsoft standards. Put most-used shortcuts on the toolbar. That does not mean using the database navigation toolbar! Use tooltips, status messages, and meaningful titles and prompts. Do not use icons on buttons. Instead use one word of text, and size the button to 14x45, and never use the flat attribute except on a toolbar. It's too easy to confuse a flat button with a display string.

James suggests that on a form with lots and lots of fields, the vital fields should all be on the first tab so the user isn't automatically taken to another tab on clicking OK. If this isn't practical, use a wizard-style form. Validate all data you can. And make sure error messages really do help, and that they don't appear to blame the user for what went wrong. Front load the dictionary with as much prompt/validation/formatting information as possible to save time and increase consistency.

Read-only fields should be done as entry fields, not strings. Use the Read-Only and Skip options, and color the text background COLOR:BtnFace. This makes the field readily identifiable as an entry field and not a string, and allows the user to select/copy/paste the data, which isn't possible with a string.

After covering application design issues, with a lot of detailed information including size

[ETC Wrapup: Sights And Sounds](#)

(Jun 6,2000)

[ETC Presentation Summary: James Fortune](#)

(Jun 6,2000)

[ETC Presentation Summaries: Bruce Johnson, Andy Stapleton](#)

(Jun 6,2000)

[June 2000 News](#)

(Jun 6,2000)

[COL Updates To Resume Shortly](#)

(Jun 6,2000)



and positioning of various controls, James reviewed the various kinds of help, including contextual, conceptual, procedural, and something else I can't make out in my notes. Contextual help includes status messages, tooltips, the "What's this" help technique (made possible using CapeSoft's EzHelp); conceptual help is something like a Help button on a browse explaining what that browse is for; procedural help is how-to help, such as the steps needed to carry out a task. James recommends that forms only have contextual help and not a help button per se.

As far as what kind of help to provide, the trend seems to be away from manuals. Consider how Microsoft does help for Windows itself.

When writing help, always use the mouse method when explaining a task. Although the keyboard shortcut method should be available, that's considered an "advanced" approach.

Aside from CapeSoft's EzHelp, James recommends ForeHelp (an authoring tool). AnswerWorks is an answer wizard which works on existing help files, and Ragazzi's standardization templates assist in making sure that help IDs have been created where necessary in your program. For tutorials, consider Lotus ScreenCam (easy to use), DemoShield, or Macromedia Director and Flash. MS CamCorder got a thumbs down, and someone in the audience pointed out that CamMedia from TechSmith creates editable tutorials, unlike ScreenCam. James uses CoolEdit 2000 for sound effects, and wasn't happy at all that the machine he used for the course had an incorrectly configured or otherwise non-functioning sound system. He did however entertain us throughout with interpretations of the sounds effects we were supposed to be hearing. The show must go on.

And his tip of the day? Tip of the Day is a complete waste of time. Nobody reads those things.

What about HTML help? For now, James is sticking to WinHelp, as there's no support for HTML help yet in Clarion (reportedly it will be in 5.5).

Resources:

- Windows 95 Help Authoring Kit (MS Press, out of print)
- GWB Help Wizard (sort of like Ragazzi's)
- [DAS QuickHelp](#) – create help from DCT
- PeaBrain's [Documentation Expert](#) – create Word doc from APP/DCT
- [Apex True Help](#) – convert Word to help file
- <http://www.accountforce.demon.co.uk/jfa/>

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

ETC Presentation Summaries

by Tom Hebenstreit, Reviews Editor

Andy Stapleton: Taking Sybase & MS-SQL to the Web

The overall arc of Andy's presentation was to take a simple application (one of the example apps that comes with Clarion), convert it from TPS files to Microsoft SQL Server 7, and then to explore various options for accessing that database via the Web.

Note that although I'll refer to TPS (TopSpeed) files when discussing the conversion process, nothing that Andy said is really specific to that file driver. In other words, everything Andy said regarding the converting an application applies equally well to any ISAM type file driver such as TopSpeed, dBase, Btrieve, Clarion DAT files and so forth.

The conversion process began with changing the file drivers for all of the files in the dictionary to MS-SQL. Andy showed how Data Modeler could be used to speed that process by allowing you to change multiple files at one time. For C5.5 Professional users, the new Dictionary Property Editor can be used in a similar fashion.

Once the drivers had been changed and the file properties set up to connect to SQL Server, Andy discussed some of the field data type conversion issues. For example, you cannot use arrays in SQL databases, and you must convert Clarion long style dates to Date type fields (at least, you must if you want to access them as dates via SQL.)

After getting the files all set up, the next step was to create the database itself. Andy was emphatic that you not create your new database tables using the built-in Clarion CREATE statement, as that method does not allow fine enough control over how the table is set up. He demonstrated using the Database Synchronizer to create an MS-SQL script containing all of the commands needed to create the tables, and then proceeded to edit the script manually to optimize it.

Another tip: Always turn off the dictionary auto create attribute for SQL tables – if your program does try to create the tables, it may wipe out all existing data.

One key concept Andy stressed is that your application and your database need to be regarded as two completely distinct entities. Unlike a "normal" TPS based application

[ETC Wrapup: Sights And Sounds](#)

(Jun 6,2000)

[ETC Presentation Summary: James Fortune](#)

(Jun 6,2000)

[ETC Presentation Summaries: Bruce Johnson, Andy Stapleton](#)

(Jun 6,2000)

[June 2000 News](#)

(Jun 6,2000)

[COL Updates To Resume Shortly](#)

(Jun 6,2000)



where nothing can happen to the data until the application is written, an SQL based system can be developed and tested extensively without a single line of Clarion code being written.

How? Remember that all data manipulation (updates, queries, etc.) are done via SQL statements, and all SQL based database systems include some kind of direct access method that let you communicate directly with the database using those same SQL statements.

For example, a utility that lets you type in SQL statements, submit them to the SQL engine and then view the results can essentially mimic all of the data-related functions of your program. And since the data base engine itself should be handling all updates, referential integrity and so forth, you can use simple SQL statements to validate everything directly within the database.

The upshot is that if you first ensure that your database is airtight, there is no way your program (or any other program that tries to access the data) can screw it up.

Inevitably, someone asked the question if they could use the same application for either TPS or SQL. Andy said while it was theoretically possible (apart from the necessity of having two dictionaries), it would basically defeat the entire purpose of using SQL. The program would treat the SQL system as if it was TPS files, i.e. do everything itself rather letting the SQL engine do the work.

His suggestion was to use dual apps and dictionaries, with the TPS app being the primary one. He would export changed procedures from the TPS based app (using selective TXA exports), import them into the SQL app and then tweak the code for SQL. A simple example might be a process procedure that loops through a number of records and updates a flag. In the SQL based system, the same result could be obtained with a single command.

After getting the app working with SQL Server, he spent the remainder of the time discussing setting up the IC/WebBuilder Application Broker and demonstrating web access. He also showed a couple of impressive applications he currently has running using WebBuilder and SQL Server.

In answer to the other question that is *always* raised when people start discussing the Application Broker (i.e., how does it scale), he said he had had 184 people actively exercising one of the web applications without any real impact on performance. He estimated a comfortable upper limit of 250 active sessions, and suggested simply adding more servers to cover more users (practical and inexpensive, given the cost of hardware nowadays).

As a bonus of sorts, Andy gave a portion of his time after lunch to Mike Gorman of [Whitemarsh Information Systems Corp.](#) Mike is (if I remember correctly) the secretary for the committee that is developing the new ANSI SQL 2000 standards, and he gave us an overview of what is currently being proposed and/or incorporated into the standard. Items include new data types such as arrays, composites, references and a whole raft more. Additionally, many new language constructs are being added to the specification.

After listening to Mike, my personal take is that they are extending SQL to point where the database basically becomes a fully self-contained programming environment as well as a data manager. About the only thing missing is an interface for the user.

If you are interested in the ANSI SQL 2000 details, the complete working documents

can be downloaded from the Whitemarsh site (along with a lot of other resources).

After Mike had finished, Andy made a salient point that people should not be holding their breath waiting for these advanced new standards to appear in actual products. As an example, he asked Mike how many of the major RDBMS vendors had fully implemented the advanced level of the ANSI 1992 SQL standards. The answer was ZERO.

Bruce Johnson: Understanding ABC So You Can Embed Your Own Code

Bruce Johnson (of CapeSoft fame) presented this extra class on the Saturday after the conference proper had ended. Let me just start by saying that this presentation is the one that all novice (and probably many intermediate) ABC users have been dying to see.

In all my studies of (and struggles with) ABC, I had never seen anyone (from TopSpeed or otherwise) present such a straightforward, lucid and just plain sensible explanation of the most basic concepts of using the ABC classes. For example, at one point Bruce put up a diagram that simply laid out how a WindowManager procedure worked, tracing how the procedure interacted with the underlying ABC class. Two columns (specific methods from the procedure on the left, base class methods on the right), some lines between them to connect the dots, and when Bruce was done explaining it, the audience actually burst into spontaneous applause (I'm not kidding). You could just see the proverbial light bulbs popping on all over the room.

Some of his key points were about the difference between *generic* and *specific* code. Generic code is the code within the ABC libraries that knows *what* to do, but not what to do it *to*. Specific code is the code generated into your procedures by the templates. That is the code that fills in the blanks for the library, e.g., specifics about this window, that control, these particular files, etc. Take a look at an Init method in one of your generated procedures and what you'll basically see is the procedure telling the library all of the details behind your choices in the procedure properties (i.e., the templates). In other words, filling in the *specific* details.

Also stressed was that virtual methods are the key to embedding your procedure *specific* code into the pre-compiled *generic* library code. To quote Bruce: "Put another way, virtual methods allow the *Library* to call the *Program*."

Apart from the nuts and bolts dissection of ABC, Bruce also provided some very useful tips, such as using the module view of source when you want to see what's really going on in your generated code. Note that I don't mean the application module view, but rather the module option on the context menu that pops up when you right click on a procedure.

How does this help? Well, like most people, I usually chose 'source' when I wanted to see the generated code, and was then presented with a *huge* (and therefore daunting) list of every possible bit of code that *could* be generated. By using the module option, you only see what was *actually* generated, and trust me, that is a whole lot easier to understand and digest. To illustrate, I just chose a fairly simple browse with no embed code from one of my applications. In source view, the editor displayed 3,387 lines of code (ACK!). In module view, it was 170 – and that included the module code surrounding the procedure as well as the procedure itself. Viewed that way, you can actually wrap your brain around it and start understanding it.

Ever pragmatic and to the point, Bruce answered some questions on why ABC didn't do

this or that with the equivalent of a shrug and a that's-the-way-it-is-so-let's-just-get-around-it. It was quite refreshing after some of the heat and bombast surrounding ABC.

While a single day class couldn't possible cover all of the nuances and usages of the ABC class library, this one did an outstanding job of establishing a broad framework for the details to be plugged into. When it was over, ABC didn't seem quite the mysterious black box that it had before – and that is a marvelous trick to pull off.

Editor's Note: Jim Haddix writes:

At one point Bruce turned away from the diagram display he was lecturing from and tried to elicit some feedback. He asked if the stuff he was talking about was valuable to us. There was a moment of silence as the attendees tried to come up with the adjectives necessary to express our appreciation. A low voice behind me responded quietly...

"Man, we're all holding our lighters up!"

Yeah...that was it exactly.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

ITEMIZE Your Parameters For A Better OMITTED()

by Carl Barnes

A big part of becoming a better programmer is adopting standards, methods, and habits that reduce the possibility of bugs getting into your code and make your code more readable. If you can get rid of a habit that leaves the door open to a bug, then you've squashed the bug before it exists. Or, from a more positive point of view, if you can standardize on methods that don't allow bugs in, then you won't need an annual contract with an exterminator.

This article will examine better ways to code omissible procedure parameters. Because such parameters are optional, it's generally wise to use the OMITTED() function to learn whether a parameter (defined by its ordinal position in the parameter list) has been passed before you try to use it. For various reasons, OMITTED() can be a bit tricky to use. At the end of this article you'll find a template that implements a standard method of handling omissible parameters in the IDE, and so eliminates bugs and saves time.

Procedure Prototypes And Declarations

Before delving into omissible parameters I need to say a word about changes to parameter lists. Prior to C4 the prototype contained only data types, and the parameter list contained only labels. Since C4 the procedure prototype and the parameter list are identical with both data type and label, except the parameter list cannot have return type information. This was required to implement procedure overloading, where the same procedure name can exist multiple times with different parameter types.

This enhancement can save you time and reduce bugs. You can enter the prototype with labels and simply copy/paste it into the parameter list, then remove the return type. Using this method will also make your code more readable since you will not have to refer back to the map when reading the procedure code. Below is an example of the old and new ways of prototyping:

[ITEMIZE Your Parameters For A Better OMITTED\(\)](#)
(Jun 14, 2000)

[The Clarion Advisor: Handling Addresses](#)
(Jun 14, 2000)

[June 2000 News](#)
(Jun 14, 2000)



```

MAP
CW20OldWay FUNCTION(STRING ,<LONG> ,<STRING> ) ,LONG END
CODE
CW20OldWay FUNCTION (Field1,Date,Field3)
MAP
C4S_New_Way PROCEDURE(STRING Field1,<LONG Date> ,<STRING
Field3>),LONG END
CODE
C4S_New_Way PROCEDURE(STRING Field1,<LONG Date> ,<
<STRING Field3>
)

```

OMITTED(): Big Mistake Or Bad Idea?

A recent thread in the newsgroups suggested that the OMITTED() function violates one of Bruce Barrington's goals for the Clarion language: the programmer should not be required to do something the compiler can easily do. Making me count the parameter number passed for use in the OMITTED() function leaves the door wide open for me to make a mistake. Worse, the compiler will not complain if the number specified points to a required parameter. Clarion OOP syntax adds another buggy wrinkle to the problem because all class methods receive the class itself as a hidden first parameter. This means you have to add one to the expected parameter number to get a valid test; OMITTED(1) will always return true when used on a class method.

Below is an excerpt of the OMITTED() example code from the Clarion Language Reference Manual (LRM).

```

SomeProc PROCEDURE(STRING Field1,<LONG Date>,<STRING Field3>)
CODE
IF OMITTED(2)      !If date was omitted
Date = TODAY() ! substitute system date END

```

As the example shows, if a procedure parameter is prototyped with angle brackets around it, then it is omissible/optional. In the above example code the Date field and Field3 are omissible, and the code IF OMITTED(2) returns True if the Date was not passed by the calling code. Refer to the Clarion LRM for more information on OMITTED().

The newsgroup thread suggests changing the syntax to OMITTED(Date) and letting the compiler figure out the ordinal number of the variable. This would be an excellent enhancement to the Clarion language. However, it could break existing code since OMITTED() currently allows a variable as its parameter. So if this is enhanced in C6 it will probably require a new syntax.

While it's nice to think about enhancing the language, Clarion version 5 exists now, and we all have to work with it. With a little help from the compiler, you can use a standard way of coding that will reduce bugs and provide more readable code than a simple OMITTED(2).

Itemize Your Parameters

Clarion version 4 added the ITEMIZE statement which allows grouping equates and letting the compiler do the numbering. This can be used to generate a list of equates for use with the OMITTED() function. In the below example code the remark on each line shows the actual equate label and value generated by the compiler.

```
Omit          ITEMIZE,PRE
Field1       EQUATE          ! Omit:Field1 EQUATE(1)
Date         EQUATE          ! Omit:Date   EQUATE(2)
Field3       EQUATE          ! Omit:Field3 EQUATE(3)
                END
```

The PRE attribute tells the compiler to take the ITEMIZE label Omit and prefix it to each equate label with a colon (and not a period). You can also specify the prefix, e.g. PRE(Omit). If you do not specify PRE or PRE(Omit) the labels will *not* have a prefix. ITEMIZE does not work like GROUP, so you can't use the dot syntax. Check out ITEMIZE in the LRM for more information. Using the above ITEMIZE equates lets the compiler do the numbering and results in more readable code shown below.

```
IF OMITTED(Omit:Date) !If date parameter was omitted
    Date = TODAY() ! substitute the system date
END
```

While there is no free lunch, this way of coding omissible parameters comes pretty cheap. It only costs compile time, as there will be no increase in size of the executable code.

Leaving Bug Doors Open

While the above ITEMIZE code looks nice, it still leaves the door open to using a OMIT equate that is not omissible. So I would never actually code it exactly that way. For any parameter label that is required I would put an "XX" or some other character string at the beginning. This way if I coded IF OMITTED(Omit:Field1) the compiler would error. The revised code is shown below.

```
Omit          ITEMIZE,PRE
xxField1     EQUATE          ! Omit:xxField1 EQUATE(1)
Date         EQUATE          ! Omit:Date   EQUATE(2)
Field3       EQUATE          ! Omit:Field3 EQUATE(3)
                END
```

The IDE Doesn't Help

Yet another bug door remains open. If you are doing this through the IDE then you enter the prototype in the procedure properties window and enter the omit equates in an embed point. These two pieces of code must be in sync or you will have bugs. Having these two pieces of code so far apart does require discipline to keep them in sync. But that's what adopting standards in coding is about. A little pain writing code yields more readable code and fewer bugs.

I can hear you saying, "That's a good method, but I'm not taking the time to type all that when I can just type IF OMITTED(2). Or maybe you're saying "I can count to two, this guy must get paid by the keystroke." Or maybe it's "I'll probably screw up the equates so why bother." These are all valid reasons; this method really is too painful to use in the IDE. In hand coded modules where the equates are right under the procedure prototype it would be fine.

A Template To Build The Equates

If a Clarion template could write all this code it would eliminate the pain of typing all the equates and any possibility of doing it wrong. I wrote such a template and it's free. The template takes the prototype and parameter list entered on the procedure properties window and parses it to create the equates for you. Simply add the extension template to your procedure and the equates will be generated for you.

This is a simple template of about 100 lines, which you may want to study if you're interested in writing your own templates. Its purpose is to parse two strings and generate some code, and it demonstrates string handling template code and how to write and call template functions. Nik Johnson helped out with the template, providing some prototype examples I had not thought of and the template code to parse them.

Below is a sample of the code the template generates. The generated code does not use the PRE attribute with the ITEMIZE so that the complete equate label exists on each line. This makes it much easier to copy and paste the equate into your code. I'm somewhat of a copy/paste maniac, so I included the IF OMITTED() in the generated code too. If you use source view you will be able to see this generated code and easily copy it into your embedded code.

```
OmitParm          ITEMIZE
NotOmit:Field1    EQUATE
Omit:Date         EQUATE ! IF OMITTED(Omit:Date)
Omit:Field3       EQUATE ! IF OMITTED(Omit:Field3)
                  END
```

The Class Wrinkle

Class method declarations add another potential bug to OMITTED(). Every method declaration has a hidden or implicit first parameter, which is the class reference (or SELF). In methods, all OMITTED() ordinal numbers need to have one added to them. The below example code shows a method declaration with an omissible parameter. While Date appears to be the second parameter the ordinal number is three.

```
MyClass.MyMethod PROCEDURE(STRING Field1,<LONG Date>,<←
  <STRING Field3>)
  CODE
  IF OMITTED(3) !If date was omitted Date = TODAY() !
  substitute system date END
```

Using the ITEMIZE method described above can improve this code. A new SELF parameter needs to be added to the equate list to take the number one slot and the rest remains the same. Below is the code written using the ITEMIZE method:

```

MyClass.MyMethod PROCEDURE (STRING Field1, <LONG Date>, ←
    <STRING Field3>)
    OmitParm
        ITEMIZE
NotOmit:Self      EQUATE
NotOmit:Field1   EQUATE
Omit:Date        EQUATE
Omit:Field3      EQUATE
        END
CODE
IF OMITTED(Omit:Date) !If date parameter was omitted
    Date = TODAY()    ! substitute the system date
END

```

There are two additional ways of coding so the equates start at the number two without having the equate for SELF. First would be to seed the ITEMIZE start value by coding it as ITEMIZE (2). Second would be to have the equate for Field1 specified as EQUATE (2) . ITEMIZE then numbers subsequent equates sequentially from two. See the LRM for more information on these features of ITEMIZE. I prefer to have the obvious SELF equate (in the example above) rather than the little tiny "2". Forced to choose between these two options I would code it as ITEMIZE (2). The EQUATE (2) could become a bug if the parameters were reordered and you forgot to move the (2) up to the new first parameter.

The Best Move Is Not To Play

If the computer from the movie War Games was presented with the OMITTED () paradox and read the LRM, it might determine that the best move is not to omit. Clarion 1.5 introduced the concept of defaulted parameters. If a parameter is omitted the default value is passed to the procedure. (It is no longer an omissible parameter and so the OMITTED () function will always return false.) The above example has been coded below using the default method to set the value of Date to zero if it is not passed:

```

MyClass.MyMethod PROCEDURE (STRING Field1, LONG Date=0, ←
    <STRING Field3>)
CODE
IF ~Date          !Date parameter was omitted (or 0)
    Date = TODAY() ! substitute the system date
END

```

This has the possible side benefit of any date specified as zero being set to today's date. The default value can only be a simple numeric or equate and not something like =TODAY () . The default method can *only* be used for numeric parameters passed by value. For all strings and parameters passed by address the <omit> method must still be used.

Omitted Variables Passed By Address

Throughout this article you have seen example code that assigns a value to the Date field even though it has been omitted. If a parameter is passed by value then the omitted variables still exist and can be assigned values and used (except for strings which are covered below). Values passed by address will *not* exist if omitted and must not be referenced. One way to handle this is at every point you use the variable you wrap the source in an IF NOT OMITTED (. . .) . If the variable is used much this could be a lot of extra code. Probably the easiest way to handle this is to create a local variable, assign it the passed value and use that in the procedure's code,

as follows:

```
SomeProc2 PROCEDURE(<*LONG InDate>)
OmitParm      ITEMIZE
Omit:InDate    EQUATE
                END
LocalDate     LIKE(InDate)

CODE
IF OMITTED(Omit:InDate)    !If date was omitted
    LocalDate = TODAY()    !      substitute system date
ELSE
    LocalDate = InDate     !      take the passed in date
END
IF LocalDate < TODAY()...
```

This method can also be used with variables passed by value, but it does add some code which isn't strictly necessary. It is the safest method to handle all omitted parameters. If the reason you are passing by address is that you want to return a value in that variable then you would not want to use a local variable as demonstrated above. You would want to handle things with a reference as show in the string examples below, or always check for the parameter being omitted before using the variable in an assignment.

Omitted Strings: A Special Case

Strings are *always* passed by address. When a string is passed by value the compiler makes a copy of the string, passes the address of that copy and when the procedure returns, it destroys the copy. So you must always write code to handle the omitted string. One easy way to code this is with an ANY variable as show below. If the string is not omitted the ANY is used as a reference to the passed string, and if the string is omitted the ANY is used as a variable length string.

```
SomeProc3 PROCEDURE(<STRING WordDate>)
OmitParm      ITEMIZE
Omit:WordDate  EQUATE
                END
LocalDate     ANY

CODE
IF OMITTED(Omit:WordDate)    !If date was omitted
    LocalDate = FORMAT(TODAY(),@d4) ! substitute system date
ELSE
    LocalDate &= WordDate      ! Local is reference to
END                            ! WordDate
```

The ANY is very handy if you do not know the approximate length of the local string you will need when a string is omitted. But an ANY is somewhat expensive and has limitations when passed to other procedures (see the LRM for details). It also cannot appear in Report or Window structure. I'm guessing this is because users can't ever find the "any key."

In the above case I know that the longest date will be "September 30, 9999" or 18 bytes. More efficient code that uses a STRING reference would be:

```
TodaysDate      STRING(18) !If Date is not passed build one
LocalDate       &STRING    !a pointer to the passed date
                                   !or Today

CODE
IF OMITTED(Omit:WordDate)           !If date was omitted
    TodaysDate = FORMAT(TODAY(),@d4) ! create a local todays
    LocalDate &= TodaysDate         ! LocalDate is ref to
                                   ! Today
ELSE
    LocalDate &= WordDate           ! Local is reference to
END                                  ! WordDate
```

Summary

These are the key things to remember:

1. Copy your prototype and into your parameter list since they can be identical.
2. Avoid using IF OMITTED(2), it's better to define equates
3. Use ITEMIZE to let the compiler number your equates
4. Use the included template to create equates in the IDE
5. Avoid omittable parameters by using default values
6. Exercise care with address passed omittable parameters, they do not exist
7. Exercise care with all omittable strings, they do not exist

[Click here](#) for a template which will create the equates automatically using the parameter list entered in the Procedure Properties.

[Carl Barnes](#) is an independent consultant working in the Chicago area. He has been using Clarion since 1990, is a member of Team TopSpeed and a TopSpeed Certified Support Professional. He is the author of the Clarion utilities CW Assistant and Clarion Source Search.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)

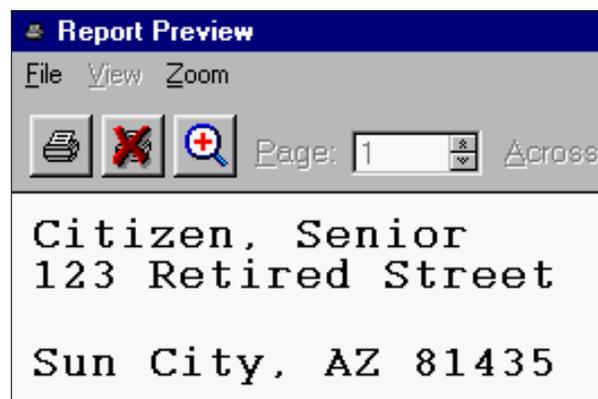
Standard Address Handling

by Steve Parker

Most of us have dealt with addresses at one time or another. It is not at all unusual for the typical address block to be composed of one or more name fields, a company name field, two address lines, city, state/province and postal code.

Usually, the company name field and address fields present problems: what if one or more is blank? What happens when these fields are printed? Figure 1 shows one possibility.

Figure 1. Standard population of address fields.



The first line shown in Figure 1 is produced with entirely standard code:

```
LOC:Name = Clip(CUS:LastName) & ', ' & Cus:FirstName
```

But the empty address line looks entirely unprofessional.

Be Gone, Blank Line

The obvious solution is to create two local variables. If there is a second address line, the first local variable will hold it and the second variable holds:

```
Clip(CUS:City) & ', ' & CUS:State & ' ' & CUS:Zip
```

If there is no second address line, this code goes into the first variable and the second is blank. So, the final code ends up looking something like:

[ITEMIZE Your Parameters For A Better OMITTED\(\)](#)
(Jun 14,2000)

[The Clarion Advisor: Handing Addresses](#)
(Jun 14,2000)

[June 2000 News](#)
(Jun 14,2000)

TopSpeed**Clarion 5**
by TopSpeed

FREE Microsoft Internet Explorer

etc
EVENT SPONSOR



```

If CUS:Address2
  LOC:Var1 = CUS:Address2
  LOC:Var2 = Clip(CUS:City) & ', ' & CUS:State |
            & ' ' & CUS:Zip
Else
  LOC:Var1 = Clip(CUS:City) & ', ' & CUS:State |
            & ' ' & CUS:Zip
  LOC:Var2 = ''
End
  
```

Effective, but an awful lot like work.

Ah, DOS, Where Are You When I Need You?

In CDD/CFD, results like this were a clear indication that the developer was new to Clarion and had neither done the tutorial nor read the manual.

Clarion for DOS gave us three important attributes: Clip, Suppress Row/Col and Suppress Line.

Clip, if checked, did the Clip() for us. And, if Clip was on, Suppress Row/Col was enabled for the rest of the fields on the line. Checking Suppress Row/Col moved the field(s) left, as far as possible.

So, in DOS, clicking these attributes had the same effect as declaring a local variable and typing the line of code shown above.

Suppress Line was even better.

Suppress Line automatically suppressed the line if the variable(s) on the line were empty. This suppression included the carriage return/line feed at the end. In other words, Suppress Line moved full lines into the proper place for you.

Not all of these options are available to us in a page-oriented report formatter. [expletive deleted]

My thanks to Russ Eggen for pointing out to me that Suppress Line is available. It's now called SKIP, but it isn't described in the report formatter control help.

The SKIP attribute can be attached to strings and text boxes by right clicking on the control and clicking SKIP. It works for columnar reports as well as labels. All controls populated on a "line" must have the attribute (well, that makes sense).

An Alternative

There is nothing inherently wrong about the two-variable solution or with the SKIP attribute. Both work. And it is hard to argue with what works.

However, I found myself in a situation where a set of same-sized variables would have made finding the address fields harder than it needed to be. An "address block" would have worked, visually, quite well.

A text field perhaps?

Figure 2. An address block in a text field.

Detail [detail]

Purchase Order: \$\$\$\$\$\$\$\$ Vendor Code: \$\$\$\$\$\$\$\$

Date: <###/##/##

Revised: <###/##/##

Phone: \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

FAX: \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

Date Required: \$\$\$\$\$\$\$\$

Total P.O. Cost: -<<<,<<<,<<##

This has some appeal. With a text field, my address area is clearly visible, well differentiated from the rest of the report or window. Further, since the variable will have to be primed (or it won't have any contents to display), I have total flexibility.

In the case at hand, only the second address line is optional. So, code like Figure 3 works just fine in ThisReport.TakeRecord (before Parent Call) embed:

Figure 3. Code to suppress blank lines.

```
GrandTotal += POH:TotalExtCost
Address = VND:Name & '<13,10>'
If VND:Address1_ <> ''
  Address = Clip(Address) & Clip(VND:Address1) & '<13,10>' |
  & Clip(VND:Address1_ & '<13,10>'
Else
  Address = Clip(Address) & Clip(VND:Address1) |
  & Clip(VND:Address1_ & '<13,10>'
End
Address = Clip(Address) & Clip(VND:City1) & ', ' & VND:State1 |
& ' ' & VND:Zip1
```

Notice, by using '<13,10>', the carriage return/line feed characters, I control where lines break. Because the entire address block is generated at once, I can conditionally include or exclude lines. If, for example, city is an optional field:

Figure 4. City-optional code.

```
Address = Clip(CUS:Name) & ' ' & CUS:LastName & '<13,10>'
If CUS:Address2
  Address = Clip(Address) & Clip(CUS:Address1) & '<13,10>' | &
  Clip(CUS:Address2) & '<13,10>' Else Address = Clip(Address) &
  Clip(CUS:Address1) & '<13,10>' End If CUS:City Address = Clip(Address)
& Clip(CUS:City) & ', ' & CUS:State | & ' ' & CUS:Zip End
```

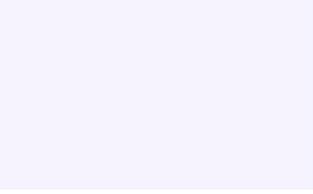
The only thing I have to do is ensure that my local variable is large enough to hold the largest combination of file variables.

Summary

Maybe I'm trapped by the old ways of thinking. Memo (text) fields used to be limited to one per file. This limit is no longer with us.

Now I now wonder whether a text field wouldn't be a better initial choice for addresses (except for postal codes; if your customer ever wants to do a bulk mailing, the postal code field must be accessible; it is the sort field of choice).

[Steve Parker](#) started his professional life as a Philosopher but now tries to imitate a Clarion developer. A former SCCA competitor, he has been known to adjust other competitor's right side mirrors - while on the track (but only while accelerating). Steve has been writing on Clarion since 1993.



Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked
Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To
Other Sites](#)[Downloads](#)
[Open Source
Project](#)
[Issues in
PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

Clarion For Zebras: Working With Barcodes - Part 1

by Simon Brewer

Barcoding is one modern technology that we've all probably avoided at one time or another, particularly in our Clarion applications. In this article I'll try to cure that aversion and demystify the use of barcodes with Clarion.

What Is A Barcode?

We've all seen barcodes, right? That series of lines appearing on just about everything we buy nowadays. Barcodes are the most widely used and enduring example of automated data capture technology invented to date.

In simple terms, barcodes are a series of carefully calculated lines (or dots) and spaces used to encode data. They are read by shining a concentrated light beam across them, decoding the reflected pattern and converting it to characters. By varying the width (and sometimes height) of the lines and spaces, barcodes can achieve very high levels of data storage density and incredibly high read accuracy.

Barcodes use different methods, known as symbologies, to encode data. Barcode symbologies are a bit akin to character fonts, and like fonts can vary greatly in size and style. As with fonts, barcode symbologies are chosen for their readability, durability and recognition factor for a given application. Common barcode symbologies are UPC/EAN (the familiar supermarket barcode), Code39 (a very common utility barcode), Interleaved 2 of 5 (a large, simple barcode for encoding numbers only) and Code128 (a very common high-density, very accurate barcode).

Some barcode symbologies are controlled to make them unique and uniform throughout the world. Perhaps the best example of this is the UPC/EAN retail barcode symbology regulated by the Uniform Code Council (UCC) in the US and Canada, and by the European Article Numbering (EAN) Associations in most other parts of the world. The control these organisations have ensure that the barcode on your can of beans does not get interpreted as a new video camera no matter where you buy it (well, at least they hope it doesn't!). As an aside, the use of these barcodes is also a core technology for

[Clarion For Zebras:
Working With Barcodes](#)
(Jun 20,2000)[True Confessions: A Tale
of Two Users](#)
(Jun 20,2000)[The Clarion Advisor: EIP
Tips](#)
(Jun 20,2000)[June 2000 News](#)
(Jun 20,2000)



Electronic Data Interchange (EDI).

As with all data storage technologies, barcodes have been continually evolving to increase the amount of data stored. The very latest symbologies can have two-dimensional arrays of lines and/or dots and can store up to 400 characters per square inch. The best known of these is probably PDF417, which all suppliers to General Motors Corporation must now use. Reading accuracy has also evolved with many symbologies incorporating complex line width, spacing and check-digit algorithms to virtually eliminate the chance of data read errors.

There are plenty of good publications available to further explain barcode symbologies. I suggest that you buy and read one or two before attempting any barcoding project.

How Do I Print Them?

Usually, the hardest part of adopting barcodes is actually creating them. I'm not talking about a whole bunch of supermarket items or books with pre-printed barcodes here, but ad-hoc barcodes with vastly varying information. Examples of this could be asset number labels, accounting documents, shipping labels, product identification labels etc. These types of barcodes typically need to be generated in-house - often from database programs - so they can be used to improve business efficiency and data accuracy.

Before I go over the different ways of creating barcodes in Clarion, I'm going to give you one very important warning: *NEVER choose a barcode symbology for your customers on the basis of how easy it is for you to implement.* It is very important for you to understand the different barcode symbologies and provide the best solution you can for your customer's needs, both now and in the future. Again, reading what you can about the different barcode symbologies and their anticipated uses is the only way to know this.

Barcode Fonts

By far the easiest way to implement barcodes in Clarion is to use a font. That's right, some barcode symbologies are so simple that they can be generated just like any other printed characters. Examples of barcode symbologies that can be printed this way are Code 3of9 (better known simply as Code 39) and the US Postal ("POSTNET") barcode.

Apart from being easy to implement, fonts are also the cheapest way to produce barcodes, with much freeware available on the Internet.

However, there is a downside. To be used in reports, fonts need to be registered with Windows. That can be quite a hindrance, particularly if you're deploying across a network. They are also very restrictive in the choice of symbologies available and it can be difficult, if not impossible, to incorporate data accuracy measures such as check digits.

Summary: Easy to implement and results produced very fast, but limited choice of symbologies, sizing constrained and difficult for large-scale deployment

PRODUCT RATING	
Implementation	
Flexibility	
Deployment	
Overall	

LEGEND	
First class all the way	
More than adequate	
Barely adequate	
Don't even think about it	

Notes regarding rating:

Implementation: How easy is this method to implement within Clarion.

Flexibility: How flexible is this method with respect to different symbologies, sizing, placement etc.

Deployment: How easily this method can be distributed with applications; number of components, registration requirements etc.

Barcode Fonts Plus API

Add API calls to barcode fonts and you have a more advanced use of font technology, and probably the most common technique in use. Basically, you place a variable barcode string a window or report with the font set to that of the chosen symbology. You make an API call to an accompanying DLL which takes the barcode data string and returns a specially formatted string. In conjunction with the chosen font this produces a complex barcode.

This is how the [Worth Data Barfonts](#), formerly on the Clarion Accessories list) and the [Stephen Mull](#) freeware templates, which utilize the WASP barcode tools from [Informatics Inc.](#), are implemented.

The downside is a moderate to high cost, and it's still messy distributing fonts. Also, if the DLL in question isn't Clarion-friendly you may have to prototype a difficult set of API calls, which could initially hinder the development cycle.

Summary: Excellent range of symbologies available, fast in operation and reasonably easy to implement. However, sizing is still constrained, potential large-scale deployment problems exist and third party template or API prototyping is required.

PRODUCT RATING	
Implementation	
Flexibility	
Deployment	
Overall	

VBX Controls

VBX controls were probably the easiest way to implement barcodes. I use the term "were" because they have been left behind as obsolete technology, and are essentially restricted to 16 bit programs only. Barcode VBXs are self-contained objects with all of the logic required for encoding barcodes and producing graphics. Basically they are/were miniature barcode engines.

A VBX control could be dropped onto any window or report and then configured on the control property window. Then just add data and stand back! A great example of this technology was the [Mabry](#) Barcode control, which I believe is still available today. Distribution involves no messy installation or registration process - simply placing the VBX control in the application directory is fine.

Summary: As I've said above, this technology is essentially dead but worth a mention for completeness.

Rating: Not applicable.

ActiveX Controls

Barcode ActiveX controls (also known as OCX controls, sometimes COM controls) are not unlike VBX controls in many respects - they are self-contained barcode engines and are very easy to implement. They are, in fact, the successors to VBX controls. Unfortunately, Clarion does not support the use of ActiveX controls on reports which is where 99% of usage is required, so they are not a valid technology to use.

If they supported output to some form of graphic image file which could then be shown in a report, they would certainly have merit and could be very useful, albeit requiring more work. Unfortunately I have not yet come across one that supports this functionality.

Summary: Forget it.

Rating: Not applicable.

Barcode Printers

There are many specialised barcode printers available containing logic to convert simple data strings to complex barcodes. Most often they are geared exclusively towards printing labels and employ thermal transfer or direct thermal printing methods. It is also possible to upgrade certain types of printer, eg. some HP LaserJets, with internal barcode processing "engines."

These devices can vary dramatically from a Clarion application's perspective, from using a proper Windows printer driver to a rather cryptic set of commands sent via the Windows generic/text printer driver.

Summary: Sometimes these devices are the best solution for a particular barcode application, but they usually come at quite a high cost and are very specific to the task.

Rating: As this method is mostly employed for very specific applications, it's hard to rate from a Clarion perspective. Generally speaking, implementation is not too hard in Clarion but other factors, such as cost and flexibility, make it undesirable if there's an alternative.

APIs

I talked above about utilising APIs in conjunction with fonts to produce barcodes. However, there are a few APIs that do not require the use of fonts. Instead, they

create a graphic image file which can easily be placed on reports.

The TAL barcode API from [TAL Technologies](#) is an excellent example of this implementation. Although there is no specific Clarion support, there is plenty of well-written documentation and excellent examples in C and VB come with it. Unusually, the license agreement for this product specifically states that API prototypes can not be divulged, so I am not at liberty to supply them here. However, even a relative novice to APIs (such as myself!) should be able to cope.

Another example recently discovered is Anex Barcode Tools from [Anex Technologies](#), but I haven't yet tried them with Clarion nor do I know of someone who has. They appear to have a significant cost advantage over the TAL equivalent and encompass similar functionality.

The TAL barcode API outputs barcodes as Windows MetaFiles (WMF) which are extremely small and fast. Because it creates an entire graphic image, it can produce extremely complex two-dimensional barcode symbologies such as PDF417 and DataMatrix. Also, the interim step of producing a graphic image means you have ultimate control over placement and sizing in reports – exact width and height can be guaranteed.

As far as distributing applications is concerned, this method only requires you to ship a small additional DLL to place in the application directory.

Summary: Fast, almost every symbology supported (including very complex 2D ones), easy to distribute and total control over barcode size. However, you must negotiate an API call and handle an interim metafile.

PRODUCT RATING	
Implementation	
Flexibility	
Deployment	
Overall	

Clarion Library And Templates

Just at the time of writing this article I happened upon a new Clarion product from Edgard Riba of [Taboga Software](#). It is an entire barcode library built with Clarion and currently in beta test for the Clarion 5 ABC templates.

This free library involves no fonts or API calls and implementation is template controlled. It supports most of the popular one-dimensional barcodes and runs at native Clarion speed – fast.

There are a few downsides. It doesn't yet support any two-dimensional barcode symbologies, there is relatively little control over barcode size, and some symbologies are missing visual details e.g. descender bars in EAN/UPC symbology. I must stress again that this template set is in beta and is very good even with these few quirks – it's well worth supporting.

Summary: Most popular symbologies supported, very easy to implement, fast, extremely easy to distribute and free, however, a few shortcomings as mentioned above.

PRODUCT RATING	
Implementation	
Flexibility	
Deployment	
Overall	

Lodestar Software's PNet

Lee White from [LodeStar Software](#) has the PNet Clarion library available which allows you to easily add US Postal Service POSTNET barcodes to Clarion applications. The cost is just \$US20 and it adds a symbology which is noticeably absent from the Taboga offering.

Rating: Not rated - limited use.

(Editor's note: Simon will have an overview of [List & Label's](#) barcoding support in Part 2 of this article.)

Reading Barcodes

So far, I've only dealt with barcode printing. There's a good reason for that; it's usually the hardest part of any barcode implementation project. However, given that you're now able to create barcodes in Clarion, there's every chance that you'll want to read them too. Fortunately that's not too hard, and I'll explain all next week.

[Simon Brewer](#) is a Senior Analyst with Email Major Appliances, Australia's largest manufacturer of whitegoods and major Clarion users. In his spare time he is also the President of the South Australian Clarion User Group and a co-organiser of the ConVic conferences.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeedFREE Microsoft
Internet Exploreretc2000
EVENT SPONSOR

True Confessions: A Tale of Two Users

by Tom Ruby

The cost of knowing what you are doing is having learned the hard way. Somehow, it doesn't matter how many books, manuals, articles or true confessions you read, you never really learn until you've made the painful mistake yourself. I confess here in the hope that you'll recognize the mistake when you make it and know the cure. In the spirit of anonymity and protecting the innocent, I will tell the stories of "User 1" and "User B."

The Tale Of User 1

I helped build a large system with many tables and display screens. Many of these tables had a field to store the initials of the employee who had created or updated the record, sent a letter, called the client or some other activity. Many tables had several of these fields. These were all three character strings. Later, I was in charge of making the second version (actually, the third), and we had a user table and surrogate keys, which we called Sequence Keys. These keys linked the users to their records by means of an arbitrary ID rather than the employee initials. I was, however, convinced by a co-worker to store the initials of the user everywhere for convenience of reporting. The reasoning went something like this: "You don't always want to be looking up the initials in the user table; it's a lot of extra work."

This system was in place, and working quite well. As time went on and the system evolved, we had lots of information linked together by these three-letter initial fields.

Then, it happened. User 1 got married. We all went to the wedding, and she was a beautiful bride. I don't know where they went on their honeymoon because I didn't want to seem nosy, but she was back at work about a week later. And her initials now were different.

To change her initials in the database we would have had to find the hundreds of places the three-letter field appeared in dozens and dozens of tables. If we just changed her initials in the user table, we would have lost the connection between her and all her work. She had two options: live with her old initials, or suffer a split personality. She decided keeping her old name was better than dealing with multiple personalities. If we

[Clarion For Zebras:
Working With Barcodes](#)
(Jun 20,2000)[True Confessions: A Tale
of Two Users](#)
(Jun 20,2000)[The Clarion Advisor: EIP
Tips](#)
(Jun 20,2000)[June 2000 News](#)
(Jun 20,2000)



Your source for
development tools
and add-ons.

Your outlet for
application sales.

had stored the Sequence Keys instead of the user's initials, we could have just updated her record in the user table and her initials would have magically updated themselves everywhere they appeared.

- Painful Lesson #1: *Always* link your tables together by the surrogate keys.
- Corollary to Painful Lesson #1: The purpose of normalization is to make sure *updates* happen properly and with reasonable IO demands, not to reduce storage space.

Now there are certain situations where you might "over normalize" your data, and you need to be careful of this. Consider a sales tracking system. You probably have an item table of some sort with a price field. You might also have a sales record table, and you might be tempted to leave the price out of this sales record because you already have the item price in the item table. DON'T DO IT! These are actually two different price fields, the normal price for the item, and the price you sold the item for. If you change the price of an item from \$2.98 to \$2.35, you probably don't mean to forget you sold it at \$2.98 to everybody you sold it to before the change.

The Saga Of User B

As the system evolved over time, I kept the painful lesson of User 1 in mind, and new development religiously followed the rule of Painful Lesson #1. Guess what? It wasn't a lot of extra work or a nuisance, always having to look up the initials in the user table. Along the line, we implemented a "file cabinet" feature in which the users could store all sorts of information related to the customers they dealt with. There was an elaborate security system whereby users could grant access to certain records in the file cabinet to other users.

Now User B was the kind of user you always like to have. He understood the system, made full use of the features, enjoyed discovering ways the computer could make him more efficient, and abused his trusted position to personal gain at the expense of the company. Ok, so he wasn't *quite* the kind of user you like to have. His activities were discovered and his employment at the company was abruptly terminated. The owner of the company called up User B's record on the user browse and pressed Delete.

Now, I can't blame the owner of the company for wanting to eliminate all memory of this villainous fellow, but what he *really* wanted to do was call up the user record and set the status to Terminated.

It was a few days before the magnitude of the owner's mistake was fully understood. Remember, User B had made effective use of the system, including the "file cabinet" feature which he used to record miscellaneous information and make it available to his co-workers. The system had very happily deleted all the file cabinet contents related to his user record.

- Painful Lesson #2: You usually *don't* want users deleting data from your database.
- Painful Lesson #2b: The owner of the company *is* a user.
- Painful Lesson #2c: The user-friendly "Are you sure you want to delete User B" message wasn't any help. Of course, the owner was sure he wanted to delete User B. The trouble was, he didn't know why he did not want to delete User B.
- Painful Lesson #2d: A comprehensive security system cannot keep users from misusing information you have entrusted them with. If you told the computer the person is trustworthy it has no choice but to believe you.

So here you have some lessons learned at my expense. Experience is a wonderful thing since it lets you recognize a mistake when you make it again.

Tom Ruby, who is no relation to the man who shot Lee Harvey Oswald, is an independent contractor living in the middle of a hayfield in Central Illinois with his wife Susan and two red-headed sons, Caleb and Ethan. He has been using Clarion for Windows since the summer of '95. Before that, he was a "TopSpeeder" using Modula II, so he has never used the DOS versions of Clarion.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
*by TopSpeed***FREE** Microsoft
Internet
Explorer**etc2000**
EVENT SPONSOR

The Clarion Advisor: Uppercasing And Capitalizing ABC Edit-In-Place

by Dave Harms

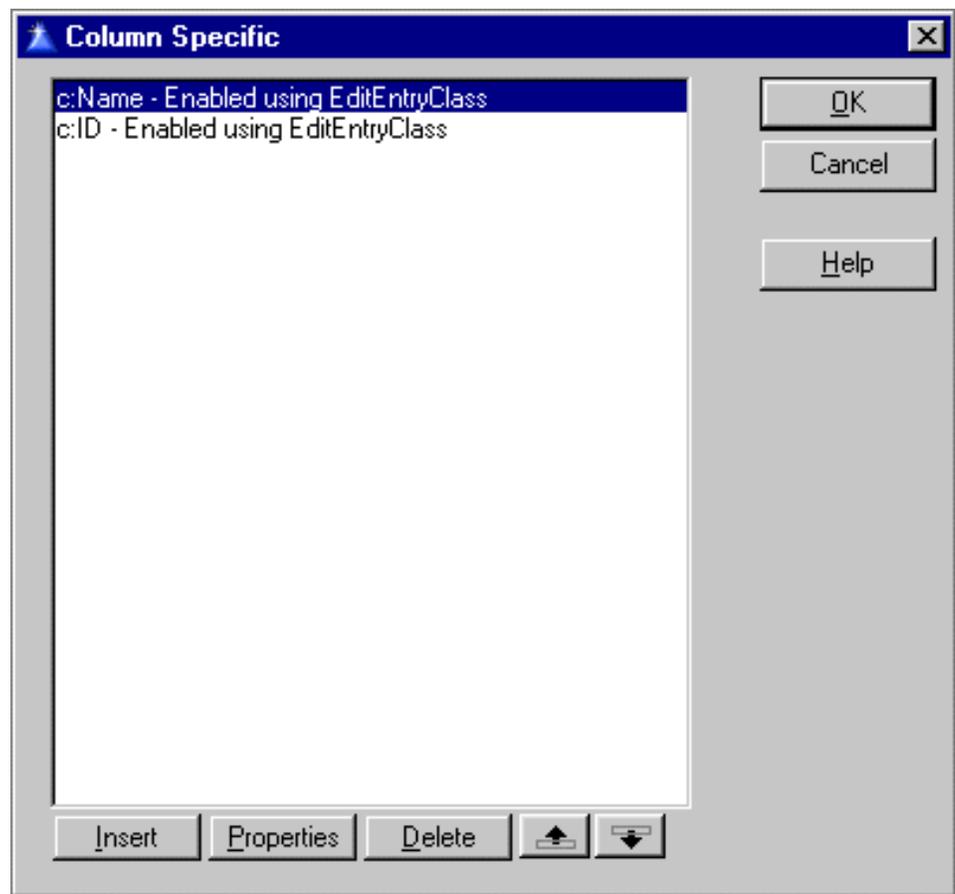
Edit-In-Place (EIP) can be an effective way to consolidate data entry onto the browse, but it doesn't always work as expected. For instance, if you have the UPR (upper case) or CAP (capitalize) attributes on an entry form, those attributes aren't carried forward to the EIP control. There's no reason why this couldn't be done in the templates, but it hasn't been, so if you want to use these attributes with EIP you'll need to embed some code.

When you enable EIP for an entire browse (by selecting the EIP checkbox on the update button prompts) the templates add the minimum code required to add basic EIP. Although you can work with this EIP manager object, it's generally easier if you let the templates create a separate object to handle each field (this also gives you full control over which fields can be edited).

Click on the Configure Edit in place button (on the procedure properties in a generated form, otherwise available from the extensions list or from the properties of any update button) to get the Configure edit in place dialog. Click on the Column Specific button at the bottom.

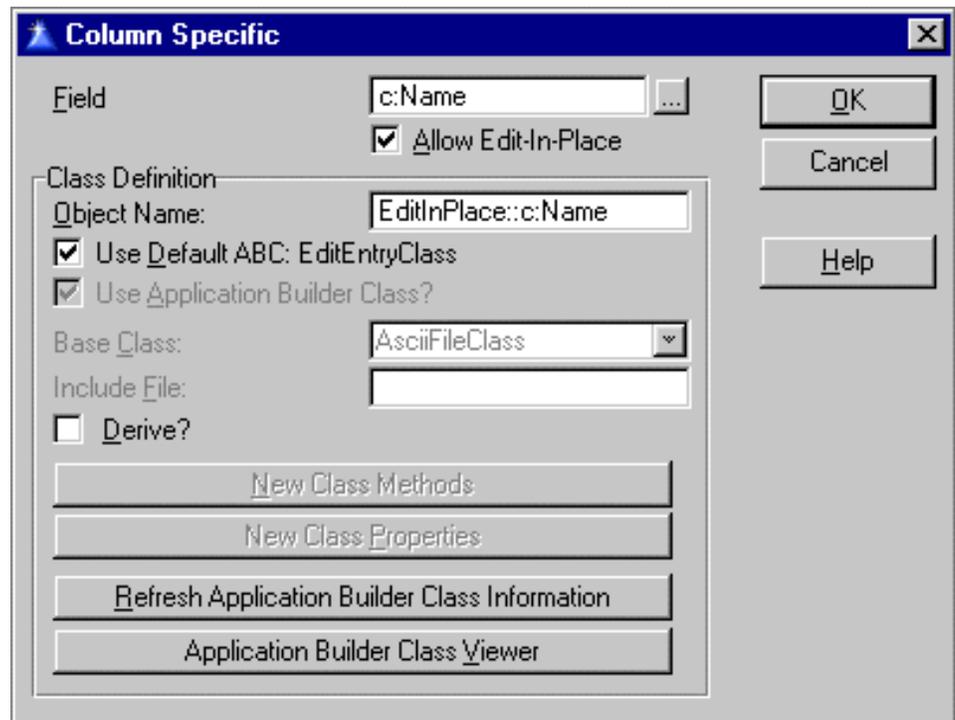
Figure 1. Creating column-specific EIP settings.

[Clarion For Zebras:
Working With Barcodes](#)
(Jun 20,2000)[True Confessions: A Tale
of Two Users](#)
(Jun 20,2000)[The Clarion Advisor: EIP
Tips](#)
(Jun 20,2000)[June 2000 News](#)
(Jun 20,2000)

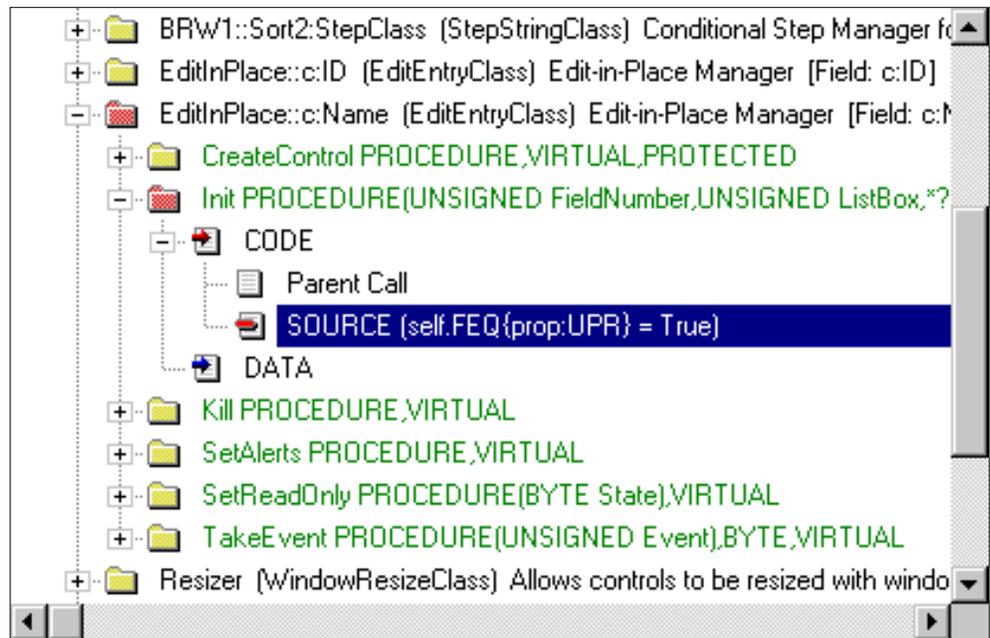


Click on Insert to add a new setting.

Figure 2. Field-specific EIP settings.



You don't actually need to do anything here other than choose the field and accept the defaults. Next, and most importantly, go to the embed list. Under local objects, find the `EditInPlace::fieldname` object and go to the after parent call embed point of the `Init` method.

Figure 3. Embedding code to force upper case.

From here it's easy. In this embed point you refer to the EIP control itself as `self.FEQ`, and use the property syntax to set the attribute. For upper case, use

```
self.FEQ{prop:UPR} = True
```

and for capitalization,

```
self.FEQ{prop:CAP} = True
```

That's all there is to it!

[David Harms](#) is an independent software developer and the co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995). He is also the editor and publisher of Clarion Magazine.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)[Subscribe
Renewals](#)[Frequently Asked
Questions](#)[Site Index](#)[Article Index](#)[Author Index](#)[Links To](#)[Other Sites](#)[Downloads](#)[Open Source](#)[Project](#)[Issues in](#)[PDF Format](#)[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

MySQL/MyODBC Update

by Dave Harms

In April of this year I wrote a series of articles on using MySQL with Clarion. Since that time a few things have changed, and I've had some additional feedback from other Clarion developers using MySQL. I also need to make a few corrections to the comments I made about MySQL during my presentation at ETC.

Errata

As I recall, I mentioned at ETC that the maximum database size in MySQL is two or four gigabytes, depending on the operating system. This is incorrect. The maximum *table* (not database) size defaults to four gigabytes, and there is no internal limit on the number of tables you can have in a database (although you can expect some performance degradation if you have thousands and thousands of tables in one directory).

The maximum table size is actually dependent on two things: the underlying MySQL file library system, and the operating system filesystem. With the most recent builds (3.23 or later) MySQL has introduced the option of using the new MyISAM file system, which is based on the ISAM library which is at the core of MySQL. Yes, this is an SQL system, but even SQL databases need to use some sort of lower level file library to actually store the data on disk.

In general, table size is limited by the operating system. On Linux-Intel, you're looking at two gigabytes per table, or four if you use Reiserfs (<http://devlinux.com/projects/reiserfs/>). On Linux-Alpha, however, you can have tables of up to *eight million terabytes!* That's 8,000,000,000,000,000,000 bytes, or four hundred million 20 gig drives. Big enough for ya?

If you're stuck at that two gig limit, you can use the merge library to treat multiple, identical tables as one table, but indexing support for merged tables has not yet been implemented.

Comparisons

If you'd like to compare MySQL features with other databases, have a look at <http://www.mysql.com/crash-me-choose.htm>. This URL generates comparisons of MySQL (current alpha), MySQL (latest stable build), Access 2000, Adabas, IBM DB2, EMPRESS, Informix, Interbase, MS SQL, MIMER, mSQL, Oracle, PostgreSQL, SOLID Server, and Sybase. Areas tested include crashability, ANSI SQL 92 types and functions, ODBC 3.0 types and functions, other types and functions, constraints and type modifiers, order by, group by, join types, string handling, various kinds of limits, and more.

Miscellaneous

[MySQL/MyODBC Update](#)
(Jun 27, 2000)[Clarion For Zebras:
Working With Barcodes
\(Part 2\)](#)
(Jun 27, 2000)[Tool Talk: Getting Into
SQL On The Cheap](#)
(Jun 27, 2000)[June 2000 News](#)
(Jun 27, 2000)



MySQL now supports transactions and replication, although this code is still in beta form (as of June 2000). Transactions can be enabled on a per table basis, which means that there's no speed penalty for tables that don't need transactions. Replication is of a fairly rudimentary sort. One server is a master, and one more other servers are slaves. All updates to the master server are propagated to the slaves automatically. You'll need MySQL version 3.23.15 or higher. You can also set up two servers such that both are master and slave, and writes to either are replicated. So far this can only be done with two servers.

Fast gets faster: you can now create in-memory temporary tables. These are useful for things like lookups where the content of the table seldom changes.

I'm not sure if anyone asked me this, but I've recently learned that there are about a million MySQL server installations worldwide.

MyODBC

To use MySQL with Clarion you need the MyODBC driver, although the June SoftVelocity newsletter indicates that native support for some Linux databases is being considered, and I would hope that includes MySQL. There is a problem with the current releases of MyODBC involving the sequence of SQL statements. Normally, the Clarion ODBC driver issues the following sequence of statements:

```
SQLPrepare
SetStmtOptions
SQLBindCol
SQLBindparameters
SQLExecute
```

MySQL, however, expects the following:

```
SQLPrepare
SetStmtOptions
SQLBindparameters
SQLExecute
SQLBindCol
```

In the alpha builds following the release of C5.5 B2, the ODBC driver can be set to a modified statement order with the following driver strings:

Driver String	Statements
/BINDCOLORDER=1	SQLPrepare SetStmtOptions SQLBindparameters SQLBindCol SQLExecute
/BINDCOLORDER=2	SQLPrepare SetStmtOptions SQLBindparameters SQLExecute SQLBindCol

The /BINDCOLORDER=2 setting creates the statement order expected by MySQL. If you're on the beta program and have the latest releases and use MySQL, please test this with the latest drivers from www.mysql.com.

Another option, if you don't mind rebuilding the MyODBC driver, is to go to <http://www.logotronic.nl/>, click on tips, and follow the example shown to patch the MyODBC driver

code. Or you can download a compiled 2.50.31 driver from <http://www.openclarion.org/cgi-bin/ftp.cgi>.

Other Driver Strings

If your browse involves more than one level of child records you'll most likely get an SQL error on the JOIN statement, because MySQL only has limited support for ODBC-style JOINS. The solution is to use the driver string:

```
/JOINTYPE=DB2
```

to force a style of JOIN that MySQL can cope with. You may also need to use the driver string

```
/USEINNERJOIN=FALSE
```

Thanks to Jim Gambon for pointing this out. There have been some changes to inner join support in recent releases of MySQL so it's possible you won't need this.

Summary

And the last word goes to Jim Gambon, who writes:

MySQL, as it is today, and for the foreseeable future, is not a Relational Database Management System. It is simply an ISAM file Database with table and key files similar to Clarion DAT files. Driving these tables is streamlined SQL engine for "atomic" database actions like "SELECT" UPDATE", etc. The tables in a MySQL Database do not know how they relate to each other. Foreign Keys are not supported to restrict or cascade changes. Changes do not "trigger" any other action within the Database to keep things like "Inventory:Quantity_on_Hand" fields correct.

"Put it on the shelf." That is what Andy Stapleton said at the ETC 2000 conference when I told him that MySQL did not currently have "triggers" or "stored procedures". I understand exactly why he would say that. Having the backend database as robust as possible before the teams of programmers and web developers start beating on it saves him (and any other Database Administrator) headaches and sleepless nights.

So why use MySQL? (And don't say "because it's free".) MySQL has no RI on the server, but it is a very fast ISAM file system. That means that the RI will have to be on the client. But the client program is our Clarion created app, and we are blessed with an abundance of data integrity tools to use in development:

Foreign Key Constraints? Tell the dictionary about the restrictions and the templates take care of the rest. Stored Procedures? No, not in the server database (yet), but the ABC classes provide fine places to embed code that needs to know about the structure of a table (FileManager) and its relations (RelationManager).

Clarion 5.5 now has File System Callbacks to be used as a sort of "client side trigger". Right before, and right after, a call to the file drivers (on ADD, PUT, SET, etc.) a procedure you define can be called. Do what you need in the Callback, and no matter where in the program the ADD or PUT happens, the proper "Quantity on Hand" update will happen. A future MySQL version will have simple stored procedures, but no triggers. You should still be able to use the Clarion CallBack (on the client) to run the MySQL stored procedure (on the server).

So this means that the same Template created RI code that works in our programs with TPS files will (or should) work when we switch to MySQL. MySQL, therefore, becomes the perfect first step away from shared ISAM files (like Clarion DAT or TPS files) in the small development shops where Clarion is often used.

So, if you have to share database maintenance chores with other groups of

programmers, then definitely "put it on the shelf." You need something with more built-in data integrity. But, if you are in charge of the programming AND the database, MySQL seems to be a fine replacement for shared-file databases like TPS.

David Harms is an independent software developer and the co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995). He is also the editor and publisher of Clarion Magazine.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

Clarion For Zebras: Working With Barcodes - Part 2

by Simon Brewer

In [last week's issue](#) I explained how to create bar codes in Clarion. Having created the codes, you'll no doubt want to read them.

Before I proceed, however, there is one more important way of creating barcodes with Clarion that I didn't cover in the first part of this saga, that is the use of a Report Writer. There is one that has quite exceptional barcode support functions and is very "Clarion friendly" named List & Label from [Solace Software](#).

The only caveat with the use of such a method is that you're obviously inserting a whole lot more than just barcodes into your Clarion app, which makes it impossible for me to rate. However, it's undoubtedly appropriate in many cases and barcodes become a "no brainer" on any report.

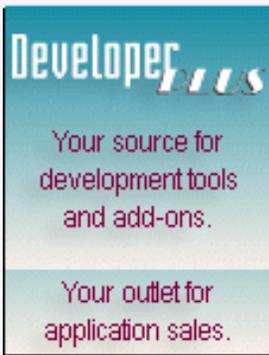
For the record, the industry de-facto standard report writer, [Crystal Reports](#), does not appear to have barcode support. I mention this as Clarion v5.5 is touted as having native support for this report writer when shipped.

Now, on with the show.....

Exactly how barcode data is transferred to Clarion programs is driven by the choice of barcode reading hardware. The choice of hardware is, in turn, driven by the intended use of the barcode. For example, a fixed barcode scanner at a supermarket checkout is an appropriate choice for serving customers who bring barcoded goods they want to buy up to the checkout. However, it would not be appropriate to unload all the contents of the supermarket shelves and march them to a checkout for the purpose of checking inventory. Similarly, it would be just as cumbersome to place the cash register and scanner on a trolley and cart it around the aisles. For this case, a portable device would be most appropriate.

Therefore, barcode scanner hardware has evolved to cater to the various physical uses of barcodes. The most common types are outlined below.

[MySQL/MyODBC Update](#)
(Jun 27,2000)[Clarion For Zebras:
Working With Barcodes
\(Part 2\)](#)
(Jun 27,2000)[Tool Talk: Getting Into
SQL On The Cheap](#)
(Jun 27,2000)[June 2000 News](#)
(Jun 27,2000)



Directly Attached "Keyboard Wedge" Scanner

The keyboard wedge is the simplest way to get barcode data into a Clarion program. A barcode scanner is attached ("wedged") in-line between the PC and the keyboard. Therefore, whatever is scanned by the barcode reader gets typed into the program just as if it's coming from the keyboard.

There are some potential difficulties with this method, the most obvious one being that of focus. When users type into a program interface they are interacting directly with it; they place the cursor in the correct field and they are looking directly at the screen – they can immediately see any feedback about what they type into it. Barcode readers are different; usually the user is more interested in finding and scanning the barcode on an object, not looking at a computer screen.

This presents some unusual challenges for us as programmers. We must strive to make our program interface as sympathetic as possible for the barcoding task. That could involve few – maybe just one – input fields on the application window, careful attention to making sure the predominant barcode input field is always selected, and highly visual or audible feedback when things don't quite go right for the user so they don't miss it.

Fortunately, most barcode readers have a number of features to help. They invariably "beep" when they successfully read a barcode, they have very sophisticated barcode decoding algorithms to make sure the data being fed in is correct and they are usually highly configurable as to how they deliver data to the host. For instance, they can typically append various keycode sequences to the end of the barcode data, such as CR+LF (the Enter key). That sure helps.

This method is certainly not doom and gloom, in fact, far from it. Keyboard wedge scanners are popular because the hardware implementation is foolproof. However, be aware that it does require good design on your part for user acceptance and smooth operation.

Serial Attached Scanner

It seems that the ubiquitous serial port just will not die, and connecting a barcode scanner to a serial (COM) port is probably the most common implementation around today. By comparison to a "keyboard wedge" scanner, most of the complication of making sure data comes in to the right place at the right time is removed; the application just needs a thread to monitor the serial port for data and process it, regardless of what the user sees (or doesn't!).

Invariably, serially connected barcode scanners allow two-way communication so an application can deliver visual or audible signals directly to the scanner device – right where the user is looking!

The downside to this type of scanner is the fact that you'll need to "get down and dirty" with serial port processing. It'll be time to dust off the old comms manuals and buy a copy of [CapeSoft's WinEvent](#), or [ClaCom](#) from [Gap Communications](#), the very long-standing third party flagship product for serial port manipulation in Clarion. Once you've done all that, it's really quite easy.

Portable Data Entry (PDE) Devices

There is an absolute myriad of devices in the Portable Data Entry category. They range

from purpose-built barcode devices to handheld PCs with attached barcode scanners. In fact, all of these devices can be thought of as acting like handheld PCs – they are computers running some form of program to collect and store barcode data. They are obviously more expensive than just buying a barcode scanner that attaches to a PC, but that is more than made up for by portability.

Generally you can't run a Clarion program on these devices, although you technically could on some. More usually, the PDE will have a simple program handling cursory validation and storing the barcode data. Later on the collected data will need to be downloaded to a Clarion database program.

Often these devices come with a serial interface allowing connection to a PC. A Clarion application could be triggered to read data from the attached device but, more usually, software is supplied with the PDE to allow extraction of data, often as an ASCII data file. It goes without saying that this is very easy to upload into a Clarion application.

There is little more I can say in this area because of the huge variety of devices and methods of collecting and transferring data to Clarion applications. I guess the golden rule is to make it as simple and as foolproof as possible to engender confidence in users.

Radio Frequency (RF) Devices

A type of device becoming more prevalent now, although quite expensive, is the Radio Frequency (RF) PDE unit. This takes the abstraction of the simple PDE device to a new level, and provides an opportunity for greater interaction with Clarion applications.

As these units are essentially PDEs, they can be running a program like any other PDE to obtain barcodes. The difference is that they are always "on-line" via radio communications and, therefore, can transfer the data they have gathered directly to a host for processing. That's where Clarion comes in. It's quite possible for the host program to be Clarion-based.

Again, it is difficult to talk about specifics because of the vast number of different implementations possible. However, there is one use of this technology that I find particularly useful. I have an application that must interface to a number of different RF PDE devices from different manufacturers via a single interface. I chose to make all of the PDE devices run a simple terminal emulation program (supplied by the manufacturers) and wrote a Clarion "Terminal Host" application.

That application uses the Windows Socket (WinSock) API to communicate using TCP/IP directly to the PDE devices. Those devices are not programmed; they simply receive all of their visual information from my Clarion application and all keyboard activity and barcode scans are delivered to the Clarion application for direct processing. This has been a particularly successful approach.

Summary

I hope that barcodes seem a whole lot easier to you now and that you can see stripes in your future. Carefully considering the three main areas will make it easy – choice of barcode symbology, creation/printing method and selection of reading devices to best suit the task. Barcoding is a very useful and achievable technology for all Clarion developers.

[Simon Brewer](#) is a Senior Analyst with Email Major Appliances, Australia's largest



manufacturer of whitegoods and major Clarion users. In his spare time he is also the President of the South Australian Clarion User Group and a co-organiser of the ConVic conferences.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)

Tool Talk: Getting Into SQL On The Cheap

By Tom Hebenstreit, [Reviews Editor](#)

Data grows. And grows.

And then it grows some more.

Sooner or later, your programs will outgrow the capabilities of the flat-file database drivers. When that happens, the solution is normally a move up to a full-fledged SQL-based Relational Database Management System (RDBMS).

Most RDBMS provide powerful features that your programs can take advantage of to both ensure greater data integrity and enhanced performance. Just as importantly, a well-designed SQL based application can keep scaling upwards with little or no further effort on your part (just throw some more hardware at it, or maybe add a parallel processing or clustering option to the RDBMS).

Even if you don't have a need yet and simply want to prepare yourself for future growth, SQL and RDBMS are the two terms you will want to keep in mind.

The problem is, Relational Database Management Systems are not only powerful, they can be expensive. Make that *very* expensive.

So how do you start getting your feet wet with SQL and RDBMS without spending a fortune?

Glad you asked.

That will be the focus of today's column – how to acquire a SQL based RDBMS at prices ranging from free (my perennial favorite) to a few hundred dollars (which is low cost in RDBMS terms).

Please note that the list is not totally inclusive, and I apologize if I've forgotten anyone's favorite RDBMS. Also, don't take the prices mentioned here as gospel; use them as the starting point for your own explorations. Prices and packages change rapidly in the software world, and who knows, you could easily find an even better deal with a bit of research.

[MySQL/MyODBC Update](#)
(Jun 27,2000)[Clarion For Zebras:
Working With Barcodes
\(Part 2\)](#)
(Jun 27,2000)[Tool Talk: Getting Into
SQL On The Cheap](#)
(Jun 27,2000)[June 2000 News](#)
(Jun 27,2000)



Microsoft SQL Server 7

SQL Server 7 is Microsoft's current flagship RDBMS. A huge improvement over 6.5, it has won over a lot of Clarion developers with new features and enhanced reliability. More importantly, it has won over quite a few IT departments, so whether you like it or not, you will quite likely be faced at some point with either the desire or the need to get familiar with it.

Paradoxically, the cheapest and most expensive way to get SQL Server 7 is via a Microsoft Developers Network (MSDN) Universal subscription (about US\$2000 annually). If your company has one of these, just grab the SQL Server 7 disks and you are good to go. You can also get some good deals on it as part of the Back Office or Back Office Small Business Server packages.

On a less harrowing level, you can download a fully functional evaluation copy from Microsoft for absolutely free. This, of course, assumes that you have a big pipe and some time on your hands, as there are twenty-two separate files totaling 217 megabytes that need to be downloaded. Oh, yeah – you'll need about a gigabyte of free space for the process of recombining the files into the installable package, etc.

A more practical alternative is to order the evaluation CD for US\$9.95. Last time I got one, it was a kit with four CDs including SQL Server 7, some info and training CDs and a bonus trial copy of Windows NT Server. No guarantee that they still give you all of that stuff, but I was impressed with the package back then.

By the way, unlike most companies that have 30-day limits on evaluation copies, Microsoft gives you a full 120 days so that you can really delve into the product.

If you own Visual Studio, the best deal is to get what used to be called the Visual Studio Plus Pack – it includes both a test SQL Server 7 license (not time limited) *and* MSDE (see the next section), along with the Windows 2000 Developers Readiness Kit and some other goodies. It may be hard to track this down now, as they rename it quite often as various bits are added and removed. When I got it, I think it was around US\$15.

Resources

MSDN: <http://msdn.microsoft.com>

SQL Server Home Page: <http://www.microsoft.com/sql>

SQL Server Evaluation page: <http://www.microsoft.com/sql/productinfo/evalcd.htm>

Special note for readers outside of the US and Canada, per the MS web site:

"This offer is available to customers in the United States and Canada only. (For locations outside North America, consult the [International SQL Server Web Sites page](#) for your country.)"

Microsoft Data Engine (MSDE)

What is MSDE? In a nutshell, it is SQL Server 7 without the administration tools. Designed for use in less demanding applications, it has a 2 gigabyte limit on database size and is not, according to Microsoft, quite as capable in a multi-user environment (after all, they don't want to cannibalize SQL Server sales.)

The bottom line, though, is that with MSDE installed on your machine you can develop

fully SQL 7 compatible applications on a single machine under Windows 9x. Since it really *is* SQL 7, everything (including RI, triggers, stored procedures, etc.) scales right over to a full-blown SQL Server system.

How do you get it? Well, if you have Office 2000, look in the \Sql\x86\Setup folder on the first disk. Note that MSDE is not installed by default – the Jet engine is (yuk!).

If you own any Visual Studio 6 component (Visual C++, Visual Basic, Visual Interdev, etc.), you can download MSDE for free from the MSDN site listed below, or order a CD for a minimal shipping and handling charge. Please be aware that this offer is available *only* for owners of at least one Visual Studio component. If you get the CD, you also get the full-blown SQL 7 itself along with a developer license for it.

Resources

MSDE Home page: <http://msdn.microsoft.com/vstudio/msde/default.asp>

Sybase SQL (SQL Anywhere Studio and Adaptive Server)

SQL Anywhere is one of the most popular RDBMS systems among Clarion developers, and with good reason. Reasonably priced, robust and filled with features that even some of the big boys don't have (or don't have working quite right), it has a vocal group of supporters on the Clarion newsgroups.

When I talk about SQL Anywhere Studio, you should know that you actually get is a whole suite of programs and tools including the server itself, Sybase Central (a management tool), SQL Modeler (database design tool), Infomaker (a report writer), PowerDynamo (a web application server) and lot more.

A 60-day evaluation copy can be downloaded free or ordered on CD for a shipping and handling charge. Weighing in at a relatively slim 78 megabytes, I'd say the download version is still a bit big for most modems.

If you think you might actually use SQL Anywhere Studio, by far the better deal is to join their free Sybase Developer's Network (SDN). Once you are a member, you can order the SBN Subscription Package. This special deal for developers includes a developer license for *all* versions of SQL Anywhere Studio (and CD's for Windows and Unix in English, plus Japanese, French and German versions). Covering one year, it also includes a year of free updates, access to developer information, more tool downloads and other goodies. And yes, Linux is included as well.

The price? US\$99 plus some shipping and handling.

I joined SBN, ordered the package and had the CDs in my hot little hands within two days. Pretty awesome, I'd say. As an aside, I thought it was pretty cool that it even came with a 230 page book – until I looked at the book and found out that it was just the license agreement (in 35 variations and over 20 languages!)

By the way, although you don't hear about it much on the Clarion Newsgroups, you can also get evaluation copies of Sybase's flagship product, Adaptive Server Enterprise, at the Sybase product evaluation site.

Resources

SQL Anywhere Studio info: <http://www.sybase.com/products/anywhere/>

SQL Product Evaluations: <http://www.sybase.com/products/download.html>

SDN SQL Anywhere home: http://sdn.sybase.com/sdn/mec/mec_home.stm

Oracle

Ah, yes - Oracle. The eight hundred pound gorilla of databases (and of downloads, as you'll see).

Oracle provides massively powerful databases (sometimes at a massively powerful price). It is their benchmarks and feature set that everyone is always trying to top, and their market share that everyone is always trying to grab.

For evaluation copies, you *could* try downloading the free 30-day trials they offer, but let me warn you that they are *BIG*. For example, Oracle 8i Standard is 450 Megs plus. We aren't talking a big pipe here; we're talking the Channel Tunnel. And that is for just one product (or more accurately, one *version* of the product).

The more reasonable deal is to order one of their CD packs that contain all of the related Oracle products for a single operating system. For example, their CD pack for Windows NT includes the Enterprise, Standard and Personal editions of Oracle 8i, along with just about every possible option you can order for them. Thus, you have 30 days to try out the exact combination of options that you need, rather than just trying the database and hoping that add-on X will solve a problem.

CD-Packs are US\$39.95 plus shipping and handling, and are offered for seven different operating systems (including Linux). Apart from the current 8i version of Oracle, they also have packs for 8.0 and 7.3. A nice touch, if you have to develop for an older version.

Stepping up, Oracle also has (surprise, surprise) a developer network that has a better deal if you need more than a trial version, but less than a mega license. Called the Oracle Technology Network (OTN), one of the benefits is the ability to purchase one-year subscriptions to what they call "Technology Tracks." Each track is geared to one operating system and group of related products.

The one of most interest to Clarion developers would probably be the Internet Servers track for the Windows NT Platform. It includes developer licenses for Oracle 8i Enterprise, Standard, Personal and Lite, as well as WebDB and the Oracle Application Server. Other tracks are available for Linux, Unix, Sun Solaris, Netware and so forth. Each track must be purchased individually.

Like the Sybase subscription offering, Oracle tracks also include free updates for the year.

The price? US\$200 per track, plus some shipping and handling. Not bad for US\$5000 dollars worth of licenses.

Side note: With Clarion 5.5 Enterprise now including a developer license for the Oracle Connect driver, it is getting easier and easier to play around in the Oracle end of the pool without your wallet getting soaked (sorry, couldn't resist that one...)

Resources

Oracle Databases main page: <http://www.oracle.com/database/>

CD-Packs ('Trials' for info or Database->CD-Packs): <http://store.oracle.com>

Oracle Technology Network home: <http://technet.oracle.com/index.htm>

IBM DB2 Universal Database

Not an RDBMS you hear about a lot on the newsgroups, DB2 is nevertheless a force to be reckoned with. If you venture into an IBM shop, chances are very good that you'll need to deal with DB2 at some point.

I couldn't find any free time-limited evaluation versions, but that doesn't mean they aren't out there (I have one for the previous DB2 version on my desk right now). What I did find during my current search is that you can download an actual developers edition (i.e., not time limited) for free or obtain it on CD for a nominal charge.

Ummm, did I happen to mention that the Oracle download was a bit on the large side? Well, hold onto your modems, because the full download for the free DB2 Personal Developer's Edition is a whopping 542 Megs spread across eight files.

If you don't feel like spending the next year downloading DB2, you can also purchase the DB2 Personal Developer's Edition at the IBM online store for US\$39 plus the ever-present shipping and handling charges.

Note: If you decide to order DB2 on CD, just follow the various links to the DB2 product list and then choose the DB2 *Personal Developer's Edition*. Ignore the "specials" link on the main page that leads to the DB2 *Universal Developer's Edition* (unless, of course, you *want* to spend US\$500.)

Resources

DB2 home page: <http://www-4.ibm.com/software/data/db2/udb/>

Trial Download: <http://www-4.ibm.com/software/data/db2/udb/downloads.html>

Ordering on CD: <http://commerce.www.ibm.com>

Pervasive SQL

Another database that has its share of vocal Clarion users, Pervasive SQL is somewhat of an odd child. Based on a hybrid system, it allows you to access data both directly (via Btrieve) and through the Pervasive.SQL engine (most RDBMS only allow access to the data through the SQL engine.)

Pervasive offers 30-day trial downloads of Pervasive.SQL 2000, and you can also order the trial versions on CD. Note that the Pervasive.SQL Server engine will not run under Windows 9x – it requires NT, Netware, Linux or a similar server OS. There is a Workgroup version, though, that should run under Win 9x.

For downloaders, Pervasive.SQL 2000 is the smallest of the bunch, weighing in at a mere 45 Megs or so.

In checking out how much it cost to order through their online store, I found out that the CD was free, but shipping and handling was US\$10. As a bonus for ordering trial software, there was a notice that I would also get \$10 off my order.

Well, after clicking "next" to get to what I figured would be some kind of order confirmation screen, all I got was a Thank You notice (uh, you mean I actually *ordered* it?). So, I can only report that there is now a totally free Pervasive.SQL CD wending its

way to my door. Not bad, not bad at all.

By the way, even though a P.SQL developer license was included in the box with Clarion 5, I've included Pervasive.SQL here because the version on that CD is now obsolete. I don't know if Pervasive.SQL will still be on the 5.5 CD (or what version it will be) but, as you can see, it is *very* easy to get.

And yes, there is a Developer Zone, although I didn't see any special deals there (that doesn't mean that there aren't any, just that *I* didn't find them).

Links to downloads, ordering and more information are all accessible on the Pervasive.SQL 2000 home page.

Resources

P.SQL 2000 home page: <http://www.pervasive.com/products/psql/>

Last Words On Licenses, Memberships, etc.

This may seem a bit redundant, but I should point out again that virtually every license obtained via any of these developer network packages is for *development* only – you cannot distribute it, use it in a production environment or install it for use at your clients.

In other words, they are giving it to you on the cheap so that people who want to use your fabulous program will also need to buy their database. The only exception is MSDE, which you can redistribute freely as long as you have a Visual Studio or Office Developer license (be sure to read the fine print, though). Even then, their ultimate aim is to have your users scale up to SQL Server.

Also, be aware that in most cases you will be required to join that particular vendor's version of a developer network. *Nobody* will let you grab these RDBMS systems without extracting at least *some* information from you.

The good news is that the basic memberships are free in all cases. You'll also find them invaluable once you start digging into the products, as they give you access to special developer oriented sites chock full of information, tips, tricks, other downloads and a whole lot more.

On the documentation side, very few of these trial offerings come with any kind of paper documentation. PDF and help files are the norm here, folks.

One final point: Be sure and check the operating system requirements for the versions you download or want to install. Some RDBMS require Windows NT/2000, while others have versions that will run happily under Windows 9x as well as other operating systems.

Happy SQLing!

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

Clarion News

June 27, 2000

[Clarion Manuals For Sale](#)

Clarion 5.5 will ship with PDFs, but for those who want paper, SoftVelocity has a limited number of 5.0 manuals still for sale. Most of these books have not changed significantly in 5.5.

[SoftVelocity To Focus On COM/DCOM](#)

Following the release of Clarion 5.5, SoftVelocity will be focusing on improved COM/DCOM support. XML/XSL support is also in the works.

[New List & Label Demo](#)

Simon Burrows has uploaded a new demonstration of List & Label version 7. This has the new RTF and HTML export formats as well as a new report layout designer and previewer. Problems with previous versions 'hanging' under Win 98 and NT have now also been fixed.

[Buggy 1.02 Available](#)

An update to the Buggy bug report tool is now available to all registered users. The trial version is also being updated.

[Open Clarion Perl/MySQL Tutorial](#)

Ron Schofield has posted a tutorial on installing the Perl MySQL DBI interface, as part of the project that uses Clarion templates to generate Perl code. Volunteers and feedback are still needed.

[FormatFlash 1.0 Released](#)

FormatFlash is a freeware template which adds a floppy disk format option to your apps. Works with all versions of Clarion from CW2 to C5.5 (legacy and ABC) - 32 bit only.

[PD One Touch Date, Time, and Scheduling Tools Update](#)

ProDomus has announced the gold release candidate of PD "One Touch" Date, Time, and Scheduling Tools (Beta 5). Changes/additions include an appointment class and template, recurrence flags, scheduling conflict check, field user options, legacy template support, various bug fixes, and documentation with examples.

[MySQL/MyODBC Update](#)
(Jun 27, 2000)[Clarion For Zebras: Working With Barcodes \(Part 2\)](#)
(Jun 27, 2000)[Tool Talk: Getting Into SQL On The Cheap](#)
(Jun 27, 2000)[June 2000 News](#)
(Jun 27, 2000)[Read The May 2000 News](#)



June 20, 2000

[Linder SFXMaker 1.0 Beta 1 Available](#)

Linder SFXMaker is a standalone utility that makes it easy to create self-extracting zip files. SFXMaker provides support for internationalization which allows the developer to customize or translate all dialogs, prompts and button captions that the end-user might see. Self-Extracting Zip archive files created by SFXMaker are fully compatible with Pkzip and WinZip. Price at release will be \$49.00. Beta password available by email.

[BackFlash 4.2 Released](#)

New features in BackFlash 4.2 include secure backups, multi-language support, format a floppy (32 bit only), and support for LS120 disks. BackFlash is now \$195 but it is still available for \$149 until June 30, 2000.

[XLIB Special Offer](#)

Alexey Solovjev's XLIB is on sale for \$45, or \$200 with source, through July 27, 2000. With XLIB you can create file, queue, and group structures at run time, create aliases and modify file structures on the fly, and more.

June 13, 2000

[Filter Wizard Template](#)

Jim Dennison has provided a filter-QBE interface based on the old Query Wizard style interface. This template builds filter statements and saves them to a TopSpeed file. Runs with C5 or C5.5 B2. Freeware.

[New IFT:HTTP Client Template and Demo](#)

Logic Central has uploaded a new IFT:HTTP Client template set and demo that fixes a bug affecting posting query strings. The templates are a free download for licensed customers. The demo is free to all.

[Managing Distributed Development](#)

Tigris is an open source software development suite for enabling internet-scale projects. Tigris will provide simple, powerful web interfaces to tools for version control, issue tracking, discussions and decision-making, automated builds, automated testing, project management and knowledge management. Products offered/hosted include Bugzilla and Scarab, CVS/CVS Web, Helm, Argo/UML and Joist (servlets and templates).

[NeoPlanet DDE Warning](#)

Richard Taylor has reported a conflict between a technique he described at DevCon using DDE servers (to detect a currently running instance of a 32 bit program) and NeoPlanet 5.1, which responds to all requests for a named DDE server. This may have consequences for other users of DDE. NeoPlanet has reported this as a bug and is working to correct the situation. The temporary fix is to right-click on the NeoPlanet system tray icon, select System Tray Options, and uncheck the bottom two boxes.

[Call Handling And Dispatch System \(CHaDS\) 2.1 Available](#)

RamaSoft Systems has announced the 2.1 release of CHaDS, which does call handling and dispatch, multi-warehouse inventory, service contract management, invoicing, and security features with auditing. New features include contract invoicing, additional reports, and a modified user interface. Single user version is US\$1500, \$250 per additional user.



[Gitano G-Buddy Discount](#)

Gitano's G-Buddy is available through the end of June for \$159, over 20% off the regular price (\$199). This price includes gold support (one year free upgrades and email support). G-Buddy's uses range from simple note keeping to complex organization and reconstruction of templates. Image, Note, and Template Buddy can be used by developers; Message and Color Buddy integrate into your applications.

[LogFlash 2.1 Released](#)

LogFlash 2.1 is now available. New features include EIP logging for Clarion 5.5 and up, multi-language support (German, Spanish, Portuguese and Danish), and control over the format of dates being rolled back.

June 6, 2000

[Princen Temporarily Stops Third Party Support](#)

Princen IT has temporarily halted sales and support activities for its Clarion third party products due to insufficient staff. This situation is expected to continue for some months. Princen's freeware products will still be available.

[Open Clarion Example App](#)

Ron Schofield has an example Perl app online, created with Clarion templates. This app demonstrates browse/form handling.

[Mail And Fax Template Update](#)

Stealth Software has announced that Joe van Niekerk, the original developer of the Mail and Fax Templates, is no longer a partner of Stealth, due to his being employed by the TopSpeed Development Center in London. Support and bug fixes are still fielded as before, however. The new support address is support@stealthsoft.co.za. A Clarion 5.5 compatible version will be available as an upgrade when Clarion 5.5 Gold ships. There will be an upgrade fee unless the templates were purchased within three months of C5.5 going gold.

[Three New Add-Ons for IFT:HTTP Server](#)

Mike Pasley has announced the upcoming release of three add-ons for IFT:HTTP Server. WebDB Jumpstart enables database access through IFT, and generates tables, forms and update procedures. It also works with forms created in editors such as Adobe GoLive, DreamWeaver, Composer, MS Notepad, etc. Requires no knowledge of HTML for basic functionality. Remote Reporter allows reports to be viewed in the browser. Also coming is WebCalendar. See the Pavilion example at Logic Central for demo (email for password and user id)

[Andy Ireland To Take Over Modulizer](#)

Andy Ireland will be taking over Joe van Niekerk's Modulizer, and will also be selling Appletizer and Appletizer Enterprise. Appletizer is the Modulizer engine with a Clarion class wrapper. Appletizer Enterprise will initially consist of an ISAPI engine for which you can write Clarion DLLs. Winsock and DCOM versions of the server to follow, and a report engine is planned.

[KnowledgeBuilder Beta Testers Wanted](#)

Savant Technology Partners) will be conducting a beta program for its first product, KnowledgeBuilder. Similar to an online knowledge base, this product is more about knowledge management. Beta testers are needed for this web enabled application.

[CapeSoft Graphing Beta](#)

CapeSoft will shortly be releasing Beta 1 of a new product called Insight Graphing. This is a tool written entirely in Clarion, which allows you to generate graphs from your data. The design goal is quick implementation, i.e. adding a graph to a program in around 60 seconds. Email for details on joining the beta program.

[Gitano Distributor Add-On](#)

Gitano Software has announced the G-RegPlus Distributor Add-On for developers who have people distribute their product and want better control over the distribution process through unlock code management.

[Gitano Gold Support Extended To All Products](#)

All Gitano products now carry Gold Support, including free upgrades and email support for one year from the date of purchase.

[GCalPro Beta 3](#)

Gitano Software's GCalPro Beta 3 is now available.

[ETC Cowboy CD Special Price](#)

George Willbanks is offering the CD of Andy Stapleton's ETC presentation to conference attendees for \$50.00 plus handling. Non-attendees can purchase for \$99.00 plus handling.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.