



Clarion magazine

Main Page

Log In

Subscribe

Open Source

Links

Mailing Lists

Advertising

Submissions

Contact Us

Site Index

ClarionMag FAQ

Download PDFs

Search ClarionMag

Volume 1, Number 4 - May 1999

Issue Index

[Freebie: How To Convert Your Database To SQL](#)

Scott Ferrett has kindly given permission to Clarion Magazine to reprint his Euro DevCon '99 presentation on converting an ISAM (TPS) database to SQL. This is essential reading for anyone considering the move to SQL.

Posted on May 3, 1999

[The Clarion Advisor On Editor Colors](#)

Bruce Wells provides some settings to make the most of the Clarion editor's customizable keyword coloring.

Posted on May 3, 1999

[Free Newsgroups For Clarion User Groups](#)

A reminder that Clarion Magazine makes free private newsgroups available to Clarion user groups for non-profit purposes. You can use your own newsgroups to make announcements, organize group projects, or assist with any other CUG activities.

Posted on May 10, 1999

[Larry Teames On Reports](#)

Larry continues with his new series on the mysterious (not to say sinister) inner workings of the report formatter.

Posted on May 10, 1999

[How ABC Handles Multiple Sort Orders \(Part II\)](#)

A need for speed (sorts) takes Steve Parker down a winding path to ABC's handling of multiple sort orders. Part 2 of 3.

Posted on May 10, 1999

[Interview With Ragnar Hellspong](#)

Ragnar Hellspong is the creator of ForKeeps, an award-winning shareware email, newsgroup, and forum message archiver created with Clarion. In this interview Ragnar explains his approach to development and talks about the success of ForKeeps.

Posted on May 10, 1999

[Clarion Links By Category](#)

Our regularly-updated list of Clarion-related web sites is organized into a number of categories such as Data Entry, Graphics, Knowledge Bases, Reporting, Software Development Tools, Security, Clarion Information, Consultants, Internet, and Web Design.

Posted on May 17, 1999

[The ABCs of OOP - Part 2](#)

This installment looks at inheritance and composition, and lays the groundwork for an understanding of virtual methods.

Posted on May 17, 1999

[David Bayliss On The FileManager](#)

David Bayliss begins his series on the FileManager class with a look at the file manager concepts and initialization code.

Posted on May 17, 1999

[The Clarion Challenge - A String Parser](#)

The first Clarion Challenge generated a lot of discussion and some very interesting Clarion code. This edition of the Challenge takes on string handling. How quickly can you parse a string?
Posted on May 17, 1999

[Clarion News - May 1999](#)

Weekly news from and for the Clarion development community.
Posted on May 25, 1999

[Demystifying The Debugger](#)

You've read "Debugging Without The Debugger." Now Russ Eggen shows that yes, you really can use the debugger to find bugs and solve coding problems.
Posted on May 25, 1999

[The Clarion Advisor - Fast ASCII](#)

A collection of tips for speeding up ASCII file access.
Posted on May 25, 1999

[DevCon Latin America Pictures](#)

Russ Eggen favors Clarion Magazine with some pictures from the well-attended DevCon Latin America '99 in Argentina.
Posted on May 25, 1999

[Beta Tests: SFX Setup Builder](#)

Clarion Magazine previews Linder Software's SFX Setup Builder install program version 3, currently in beta 4. Setup Builder creates compact install programs with some very nice features for Clarion developers.
Posted on May 25, 1999

Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Feature Article

The Novice's Corner: Designing A Database

By Dave Harms

In the first two articles in this series I examined the [underlying philosophy of developing software in Clarion](#), and reviewed the all-important [role of Topspeed's template technology](#) in creating applications with a minimum of coding.

All of that background information is, I think, vital to being productive with Clarion. On the other hand proficiency in Clarion is only part of the equation. You may understand all there is to know about the development environment, and you may be able to write Clarion code so pure it'll bring tears to David Bayliss's eyes, but that isn't enough to write a great application.

Software development is a bit like building a house. Many traditional programming tools correspond to hammers, nailers, saws, drills, and other power tools. In this analogy the tools help you shape the raw material into a structure. Newer software development tools, including Clarion, do the software equivalent of prefabricating large sections of a building. And very sophisticated tools like the Wizards and Wizatrons let you create the equivalent of an entire building with a certain look and feel.

This sort of technology is a great boon to many developers. Still, even the Wizatrons can't (yet) read your mind and know what kind of an application you'll want. They work on the basis of the application style you choose (or create) and the data dictionary you create (or use, if one already exists).

From the Wizatron's perspective application creation seems quite simple. All the Wizatron needs to know is the expected appearance and behaviour of the application and the data structures. Simple, right? All the developer has to do is choose how the application should look, how it should work, and what data it should store, and writing software becomes (in theory) a license to print money.

This is, of course, the reason the world still needs software developers as much as it needs software development tools. And as tempting as Wizzy technology is, there's still a lot of work for the developer to do before any given application development process hits its automation phase.

Begin At The Beginning

If you've accepted the basic assumptions of the Clarion way of building applications (MDI frame and windows, the browse/form paradigm), then you're probably not going to quibble with the fundamentals of your application's appearance. You are going to be concerned about the data dictionary (which in Clarion will determine a lot of the application's behaviour). Therefore much depends on the quality of the database design.

But how do you arrive at your database design? Unless you're into

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

If you're interested
take our poll &
let us know!

randomware, you begin with some sort of requirement, whether that's a formal document or just some half-baked ideas rattling around in your head. From this requirement you can begin to formulate some possible approaches. The possibilities you come up with will depend a lot on your past experience with application development, your familiarity with existing, similar applications, and your knowledge of what options Clarion makes available.

Application design, which ultimately includes database design, is itself a rather large subject and something I hope to touch on in future articles. I'm partial to something called "use case" analysis, but there are any number of techniques you can use to try to flesh out what the application will need to do. And you can be sure that whatever the initial requirement describes, the final product will be something more complex.

A Student Tracking Application

Somewhat irresponsibly I'm going to leave aside the principles and techniques of overall application design and focus on the data dictionary. My assumption is that you're comfortable with the kinds of applications Clarion creates by default (MDI, browse/form), and that the user's requirements very conveniently do not deviate in any significant way from the functionality provided by Clarion 5 with the ABC templates.

Imagine an application which will be used to track university students and the courses for which they register. The application should be able to generate reports showing students in a given course and the courses taken by a given student, and should provide standard student information such as address and phone number.

It's vital in any development effort to determine how to store the data in the most efficient, reliable, and useful way. As of this writing, for Clarion developers, this means using some sort of relational database. (If you're thinking about object-oriented databases, or object-relational databases, that's still in the future for us and for a lot of other developers too.)

Relational Databases

Clarion is built around the concept of the relational database. As you might guess from the name, a relational database deals with groups of information which are related in specific ways. These groups are typically called tables if you're dealing with an SQL database, or files (at least in the Clarion world) if you're using a non-SQL database. For the rest of this article I'll use the term file or data file to describe such a group, since I'll be creating a Topspeed database.

Each data file can contain zero to many records (rows in SQL parlance) of information. For instance a file of employee information would contain one record for each employee, and within that record there will be individual fields, such as one for name, another for telephone number, and so on.

One key rule of relational databases is that each record in a table must have a field which is a unique identifier. In Clarion this value is usually defined as a LONG integer and its value is typically created automatically (by the ABC class library or by code generated by the templates). This field should also not normally be available to the user to change.

Another of the rules which describe good relational database design says that data duplication must be kept to an absolute minimum. This rule is closely related to the first rule, as you will see.

Armed with these two rules you can avoid a lot of problems in your database design.

In the student tracking application you might begin with a table to hold student information. Fields could include `FirstName`, `MiddleName`, `LastName`, as well as fields for the address and telephone numbers.

Before your rush to the dictionary editor and begin creating the `Students` file (or any other file) ask yourself if you really know the extent of the data. Will the fields you propose be sufficient? Will each

student, for instance, have only one address? University students are notoriously mobile. My personal best was 13 addresses in three years (and no, I wasn't running from the law). Often students will have a permanent or home address and a residence address, and perhaps a third address when away on a work term. If you don't have the familiarity with the data to know these sorts of things then you must rely heavily on those who do. This can be trickier than it sounds because oftentimes people who are familiar with the data will omit details that to them are second nature but non-obvious to anyone else.

Avoiding Data Duplication

Clearly you'll need to store more than one address per student. If you know the exact number of addresses, you can simply add extra fields to the file. But if some students will have one address, and others two then there will be a lot of wasted space in the database. And if some students have more than two addresses you won't be able to accommodate the extra information.

Some developers use arrayed fields in these kinds of situations. Arrays are a convenient solution from a coding perspective because you can use a consistent set of labels when processing the data. All you need to do is change the subscript. But arrays don't solve the problem of wasted space, and they tend to introduce a lot of other problems when it comes to sorting data or using the fields in list boxes. Arrays do have a place, but they should be used only when they are the best solution.

In the case of student addresses, a better solution is to have a separate address file, and link it to the `Students` record by using a unique identifier.

Listing 1 shows record structures for the student and address files as created in the data dictionary.

```

Students          FILE, DRIVER( 'TOPSPEED' ), PRE( STU )
                  , CREATE, BINDABLE, THREAD
StudentIDKey      KEY( STU:StudentID ), NOCASE, OPT, PRIMARY
Record            RECORD, PRE( )
FirstName         STRING( 30 )
MiddleName        STRING( 30 )
LastName          STRING( 30 )
StudentID         LONG
                  END
                  END

Addresses         FILE, DRIVER( 'TOPSPEED' ), PRE( ADD )
                  , CREATE, BINDABLE, THREAD
AddressIDKey      KEY( ADD:AddressID ), NOCASE, OPT, PRIMARY
StudentIDKey      KEY( ADD:StudentID ), DUP, NOCASE, OPT
Record            RECORD, PRE( )
AddressID         LONG
StudentID         LONG
Address1          STRING( 30 )
Address2          STRING( 30 )
City              STRING( 30 )
StateProv         STRING( 2 )
                  END
                  END

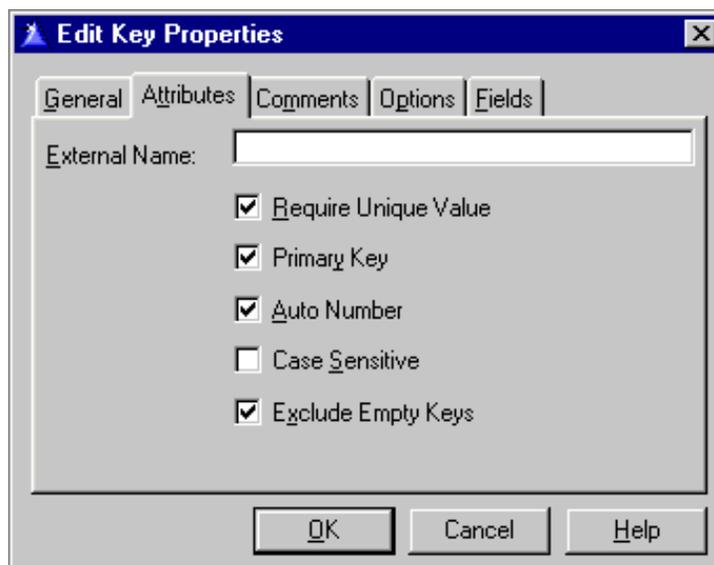
```

You can see that both files have a field called `StudentID` which is a `LONG`. This field serves two purposes. It uniquely identifies each student record, and it also provides a link between the student record and one or more address records.

TIP: For fields that you'll be using in more than one file, create a file in the data dictionary of type "Pool" and enter the base field types there. When you need a field of a type you've defined, click on the button to the right of the Derived From field in the New Field Properties window and choose from the Pool file. Do this before you enter a value in the Field Name field or you'll also have to reenter the Prompt Text and Column Heading values.

Simply having a linking field isn't sufficient. You also need to declare a key on the linking field in each file so that the linked record can be quickly retrieved. In the `Students` file the `StudentIDKey` is a unique key (it lacks the `DUP` attribute). You set this option on the Key Properties dialog, on the Attributes tab, as shown in Figure 1.

Figure 1. The key properties Attribute tab for `StudentIDKey` in the `Students` file.



`StudentIDKey` in the `Students` file has several other attributes. It's a **primary key**, which means that the key is unique and contains a value for every record in the databases. You cannot have a null value in a field that is in a primary key. The primary key ensures that there is always a way to retrieve a specific record in a file. This is essential in an SQL environment where there is normally no equivalent to a record number as there is in a flat file database.

This key also has the `Auto Number` box checked. Checking `Auto Number` won't make any difference to the way the file is declared, but it will cause code to be generated that creates a unique sequential number for the key value for every record added to this table.

(Note: QuickLoad now has an `Auto-Increment Key` option that will create an appropriate auto-numbering key.)

`StudentIDKey` in the `Address` file is slightly different. Here the `StudentID` field will contain a value obtained from the `Students` file, so rather than a primary key this is a foreign key. The `Address` file does have its own primary key in `AddressIDKey`. In an SQL database every file (table) should have a primary key.

Just as creating fields doesn't automatically give you a key, creating keys doesn't automatically give you a link between files. This has to be defined in the Data Dictionary so that the appropriate code can be generated to manage the relationship. In the case of an SQL database, you have the option of also defining the relationships in the database

itself in which case the database server can enforce any constraints. In either case, you should have the relationship defined in the dictionary.

This particular link on `StudentID` is a one-to-many relationship between the student and address records. Any one student can have from one (or I suppose zero) or more addresses.

To define this relationship highlight the `Students` file in the dictionary main window and click on the `Add Relation` button. The `Relationship properties` window appears, and it has one field filled in, `Relationship to Students`, which defaults to `1:Many`. (The only other option is `Many:1` which you could choose if you wished to create or edit the relationship from the perspective of the `Addresses` file.)

Choose `StudentIDKey` as the primary key, `Addresses` as the related file, and `StudentIDKey` as the foreign key.

You'll see the key fields listed under `Field Mapping`. You can choose to automatically map the fields by name or order (either will work in this case) or you can double-click on the individual fields to specify which field is the linking field. This is more of an issue when creating relationships with multi-component keys.

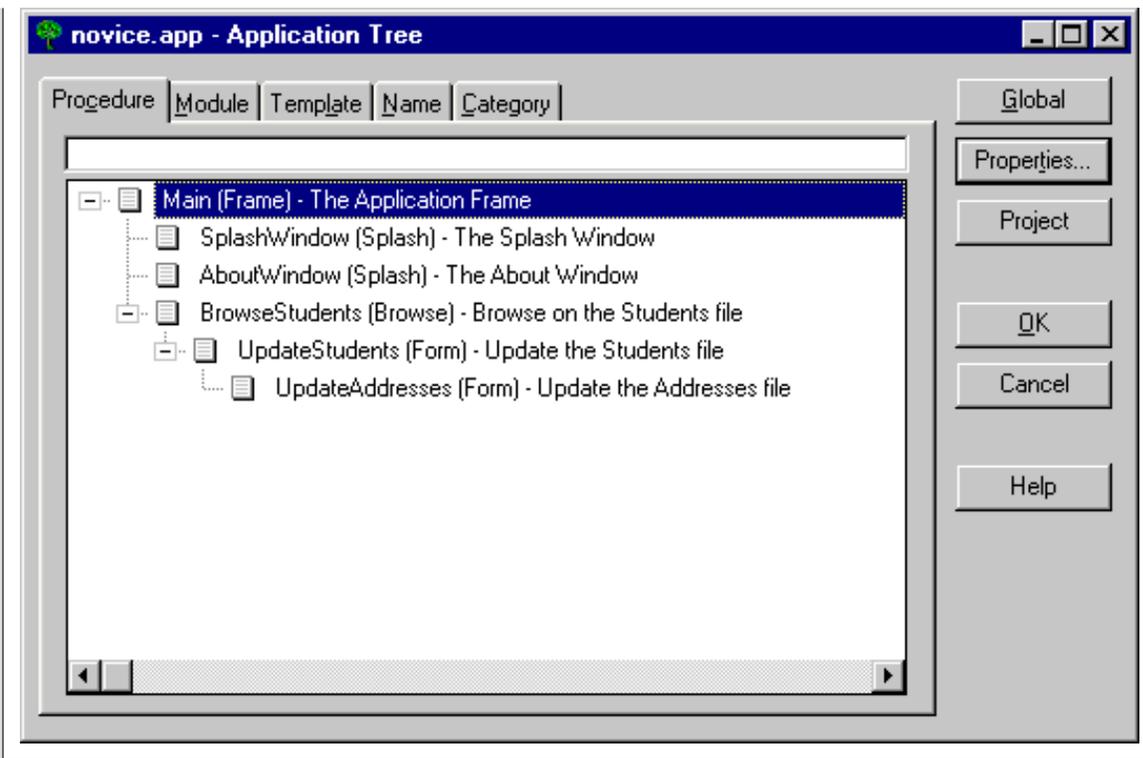
All that's left is to choose update and delete constraints, which determine whether certain actions involving related files will be allowed. The options are `No Action`, `Restrict`, `Cascade`, and `Clear`. The last three options are repeated with the annotation `(Server)` meaning that the restrictions exist but will be managed by the SQL database server rather than by Clarion code.

In a `Restrict` constraint you may not change or delete a primary key value if a related record exists. In a `Cascade` constraint any change to the primary key value will result in the change being made in any related records, and on a delete the related records will also be deleted. A `Clear` constraint clears any foreign key values on a change or delete of the primary.

In general you don't want to give your users access to linking fields anyway, so `Update` constraints shouldn't be a major issue. `Delete` constraints require more thought, and usually the choice is between `Restrict` and `Cascade`. If someone deletes a student record you're probably going to want to delete all the address records as well.

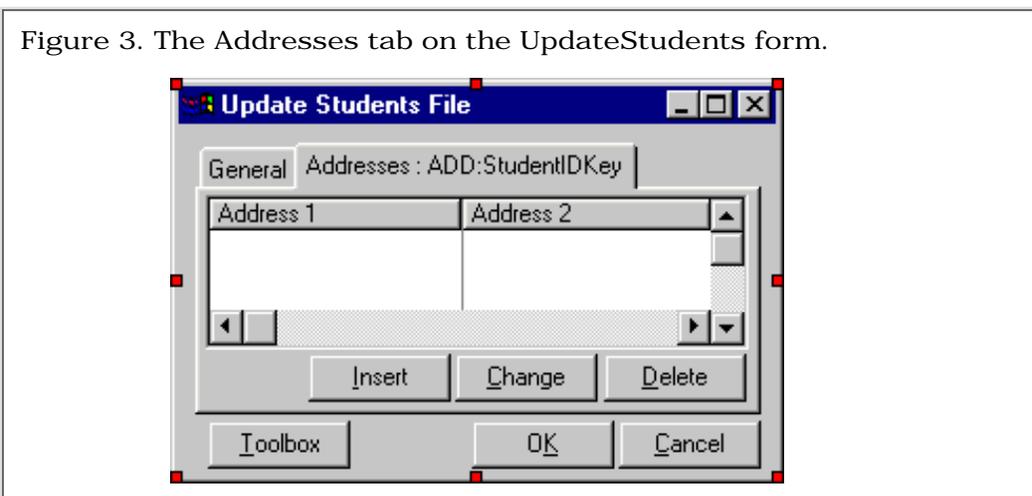
Now that you've determined the relationship you can go ahead and generate an application using the `Application Wizard` or `Wizatron`. Choose only the `Students` file when asked which files you want to be able to browse. Figure 2 shows the resulting application tree.

Figure 2. An application which browses the `Students` file.



Note that although you only specified that you wanted to browse the `Students` file Clarion has created an `UpdateAddresses` form, called from the `UpdateStudents` form. On the `UpdateAddresses` form, on the second tab, you'll see a browse for the `Addresses` file, as shown in Figure 3.

Figure 3. The `Addresses` tab on the `UpdateStudents` form.



The `Addresses` browse exists only because of the relationship defined in the `Data Dictionary` between `Students` and `Addresses`. The `Wizard` or `Wizatron` simply populated a browse control template on the tab and set it to show only those address records which are linked by `StudentID` to the current student record.

Although Clarion by default puts the address browse on the student form, you can also place an address form on the student browse window if you prefer. You'll have to do this manually as the templates don't have this option. I'll have more about this in next month's article.

A Course, Of Course

This application now has a minimal capability of handling student information. The other main requirement is to track which courses students take.

It's easy enough to define a record to represent a course. You might want fields such as `CourseName`, `StartDate`, `Instructor`, `Location`, and so forth. But think about the actual data. You have many students and many courses. In other words this is a many-to-many relationship, not a one-to-many relationship. How can you represent this information in the most compact way possible? And does the Data Dictionary support many-to-many relationships? [Tune in next month to find out!](#)

[Download the source code](#)

[David Harms](#) is an independent software developer and the co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995). His company, CoveComm Inc, publishes Clarion Magazine.

Copyright © 1999 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.



Clarion magazine

[Main Page](#)
[Log In](#)
[Subscribe](#)
[Open Source](#)
[Links](#)
[Mailing Lists](#)
[Advertising](#)
[Submissions](#)
[Contact Us](#)
[Site Index](#)
[ClarionMag FAQ](#)
[Download PDFs](#)
[Search ClarionMag](#)

Feature Article

Converting ISAM (TPS) Files To SQL Files

by Scott Ferrett

This document is based on the talk given by Scott Ferrett at Euro Devcon '99 in Amsterdam on 22 April 99. It is based on the facilities available in Clarion 5a Enterprise Edition. Later versions of Clarion may require different (hopefully less) work. However, the information on file structure changes will probably still apply.

Reprinted in Clarion Magazine with permission.

Throughout this article I will refer to TPS files. However, this article applies equally to any ISAM file format (Clarion, dBase, or Btrieve). I will refer to SQLAnywhere as the SQL driver. However, this can be replaced with Oracle, MSSQL, Scalable, AS400, ODBC or any other SQL file driver.

There are two things to do when creating an SQL database based on an existing set of TPS files: convert the data definitions and convert the data.

Converting the Data Definitions (Creating a new DCT)

The first thing to do is create a new DCT.

You then need to move the existing TPS tables into the new DCT and convert them to SQL tables. This can be done in two ways: Copy the old DCT to the new DCT then change each table one at a time; or Export the old DCT to TXD, edit the TXD and import the TXD into the new DCT.

I will use the second technique as it allows for bulk changes. However, this system is much more dangerous as you have no tools to assist you in making certain that everything you do is consistent and correct.

Driver Name

This is the most obvious change. You need to change 'TopSpeed' to 'SQLAnywhere'. Do not do this yet.

OWNER

All SQL tables require an OWNER attribute that indicates how to connect to the database where the tables live. THIS SHOULD ALWAYS BE A VARIABLE.

If you have any tables that already have an OWNER attribute, remove these.

Once you have no tables with OWNER attributes you can replace "TopSpeed') " with " 'SQLAnywhere'),OWNER (GLO:Owner) "

Key Component STRING => CSTRING

In most SQL databases trailing spaces are important when testing if a

field is equal. So 'Smith' ~= 'Smith '. This is normally only important on key components as they are used in relational links and filters. If you use CSTRINGS instead of STRINGS, then Clarion treats trailing spaces in the same manner as the SQL system.

MEMO => STRING

SQL drivers do not support MEMO fields. These can be represented as very large strings. As you will be developing 32bit applications you do not have a 64K record limit to worry about.

LONG => DATE and TIME (sometimes)

In older systems date and time data is stored as a LONG. These should be converted to DATE and TIME fields so your data will be easily accessible from non-Clarion programs such as third party report writers.

No RECLAIM attribute

SQL drivers do not support this attribute.

Remove NOCASE

Every SQL system (except P.SQL) only supports the concept of case sensitive or case insensitive keys across the entire database. You cannot specify one key as case sensitive and another one as case insensitive.

Because of this global setting, SQL systems do not require the NOCASE attribute. In fact, specifying it can SIGNIFICANTLY impact on performance.

Change Referential Integrity to Server based

This is something you get for free. You do not have to learn any SQL to get the advantage of having the server do your referential integrity. To do this you need to change all your relational constraints to the equivalent server based relational constraints. If you are editing the TXD you do this by changing;

CASCADE to CASCADE_SERVER;

RESTRICT to RESTRICT_SERVER;

and CLEAR to CLEAR_SERVER

You have now done all the quick TXD based changes. You can do the following changes in the TXD or after importing the TXD. I recommend the later.

Make Sure All Files Have A Unique Key

There is no hidden record number in SQL. So for the driver to be able to update a record you must tell the driver how to uniquely identify a record. To do this you must have at least one unique key defined for a file. Even one record control files. The driver does not know that there is only one record.

Do Not Use GROUP IDs In Keys

It is fine to use fields that are within a group as components of a key. But you cannot use the GROUP field in a key. If you have keys that use a GROUP field, you will need to change the key to use each field within the key.

Converting The Data Definitions (Create an SQL Script)

To create an SQL Script you will need two dictionaries. One is the one you created in the earlier steps. The other is an SQL dictionary. You get an SQL dictionary by either creating a new SQL database or using an existing one.

Run The Dictionary Synchronizer To Create An SQL Script

Run the synchroniser. Select your other dictionary to be the SQL

database you want to create the SQL tables in.

Set the Source DCT as the Clarion DCT you created in the previous section and the destination DCT as your SQL database.

Once you get to the synchroniser screen you need to copy all the files to your SQL database. The easiest way to do this is to highlight the top line. Press the right mouse button and select add.

Run the script

To run the script you need to run your SQL's SQL executor. Load the script and run it.

Converting The Data

Create the Conversion Program

To create a conversion program you select Create Conversion Program from the File menu whilst you are in the dictionary editor.

You then end up back in the synchroniser. Don't panic. This is where you are meant to be. The synchroniser has many faces. One of these is to allow conversion programs to be created.

Choosing your dictionaries

To create a conversion program you need two DCTs. The original DCT that contains all the TPS files and the new DCT that contains the SQL table definitions.

The most confusing part of creating a conversion program is the screen that asks you which DCT is the source and which is the destination.

The Source DCT is the one with your SQL tables in it

Your Destination DCT is the one with the TPS tables in it

Once you get to the synchroniser screen you need to copy all the files. The easiest way to do this is to highlight the top line. Press the right mouse button and select copy.

Press OK.

You now get the next confusing part of the conversion program creator. You find yourself back in the dictionary editor.

The system has actually done what you wanted. It just didn't tell you.

Edit the Conversion Program

A project convert.prj was created in the previous step.

Load this as the current project.

The first thing to do is change the properties of the project to create a 32bit program.

You will then need to edit the conversion program. The program is an OO program designed to convert an existing set of ISAM tables to a new version of those tables. As such, it does not handle converting ISAM to SQL without a few modifications. Being an OO program it allows us to make these modifications without having to hack the base code.

The file that needs to be edited is C5CVT__1.CLW.

Standard SQL Code Additions To Conversion Program

There are a few standard settings that need to be overwritten for all SQL tables. So the easiest way to do this is to create an SQLDestTable class derived from DestTable. You then change all your DestTable derived classes to be SQLDestTable derived classes.

```

SQLDestTable      CLASS(DestTable)
CreateTable        PROCEDURE ( ),RCCODE,DERIVED
AskName            PROCEDURE (BOOL _MustExist, ),BYTE,PROC,DERIVED
BuildKeys          PROCEDURE ( ),RCCODE,DERIVED
                  END
SQLDestTable.CreateTable PROCEDURE( )
  CODE
  RETURN RC:Ok

SQLDestTable.AskName  PROCEDURE (BOOL _MustExist, )
  CODE
  SELF.FileName = SELF.Label
  RETURN RC:OK

SQLDestTable.BuildKeys PROCEDURE ( )
  CODE
  RETURN RC:Ok

```

Setting the OWNER

The conversion program creates an owner variable for every table. This is a pain. You need to change every OWNER() attribute to refer to just one string. You can then either add code to get a user ID and password from the user, or hard code it.

Set Tasks

The conversion program generator does not set the right tasks to be performed when doing this conversion. So you need to edit the "SELF.Task = " line to be:

```

SELF.Task = TASK:DefaultSQL + TASK:OpenSrc - |
          TASK:Backup + TASK:UpdateDest

```

There is one of these lines for each file being converted.

Convert Does Not Handle MEMO=>STRING

There is a bug in the conversion generator where it does not generate the necessary code to convert memos to strings. So in each table that had a memo you will need to edit the Assign procedure and add the line

```
NewFile.StrField = OldFile.MemoField
```

Modifying Your Applications

The only thing you will need to do is add a logon screen to the start of your program. You will need to ask the user for a User ID and Password.

You can hide this from the user by having a hard coded User ID and Password. Even in this case you will want to add a dialog indicating the program is connecting to the server as this can take some time (up to 20 seconds is not unusual).

The connection to the server is done when you first open a file. So you need to either open a file in your Logon screen, or on the frame. You can close the file immediately. The connection will be kept open until the application terminates or PROP:Disconnect is called.

Doing This For Client Data

This article has been written from the viewpoint of the developer having access to the database. If you want to convert an existing customer's data to an SQL system you will need to:

- Get the SQL system installed.
 - Either copy over an empty database that already contains the table definitions, or create a new DCT and run the creation script.
 - Run the conversion program
-



Clarion magazine

- Main Page
- Log In
- Subscribe
- Open Source
- Links
- Mailing Lists
- Advertising
- Submissions
- Contact Us
- Site Index
- ClarionMag FAQ
- Download PDFs
- Search ClarionMag

The Clarion Advisor

Customizing Clarion 5 Editor Colors

by Bruce W. Wells

Have you ever explored the colors setup in the Clarion 5 Editor configuration? If not, then you are missing one of the most powerful features of the editor.

Imagine you're typing code, putting together a MESSAGE () statement, and you completely miss the fact that you forgot to insert one of the necessary single quotes for the message text string. You save your work, generate and start the compile only to have the compiler burp on the unterminated string. Well it's an easy fix and doesn't take too long, but how many times are you going to do this?

With your colors properly configured, the text will change colors as you type the characters, indicating proper syntax.

I use a black background in my editor because this provides the most contrast with a wider variety of colors. Bright cyan or yellow on white is nearly illegible, but on black these colors are very noticeable.

If you're interested in trying this feature and can live with my color selections, consider the following recommendations. Figure 1 shows the Colors tab on the Editor Options window (choose Setup|Editor Options). If you've made any changes to your editor colors beforehand be sure to save a copy of c5edt.ini. If you haven't made any changes you can always click on Restore Defaults if your choices don't work out.

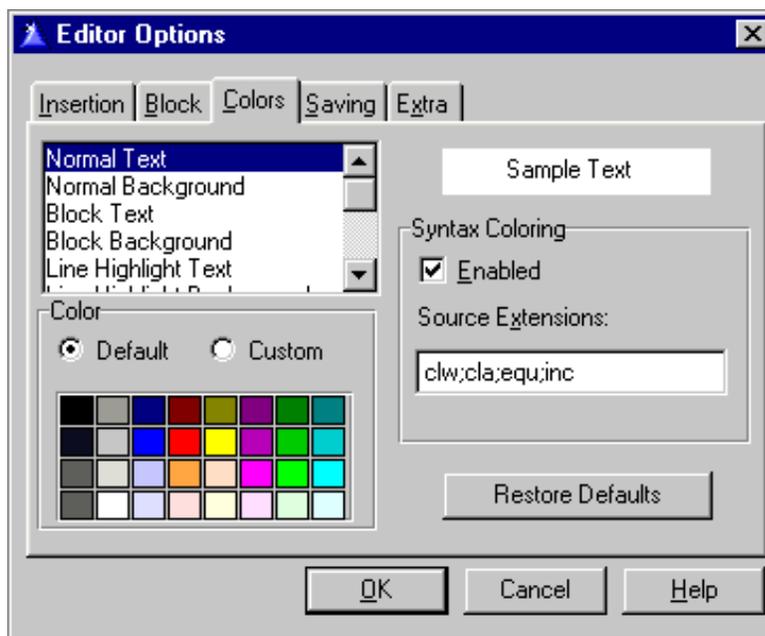
BKO
Enterprises, Inc.

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

etc
2000

If you're interested
take our poll &
let us know!

Figure 1. The Editor Options Window



Notice that the color palette display consists of four rows and eight columns. For the purpose of this discussion, I will refer to the colors by their coordinates. For example, black will be 1,1 (row 1, column 1). Below are all the components affected by color and their correspondingly assigned colors.

Normal Text	2,2
Normal Background	1,1
Block Text	4,2
Block Background	2,3
Line HighLight Text	2,2
Line HighLight Background	1,1
Error Line Text	4,2
Error Line Background	2,4
Label	3,6
Comment	2,5
String	2,7
User Identifier	4,4
Integer Constant	3,4
Real Constant	1,7
Picture String	2,6
Disabled Text	4,1
Disabled Text Background	1,1
Clarion Keywords	1,8
Compiler Directives	1,7
Built-In Procedures/Functions	3,3
Structures Data Types	2,6
Attributes	3,5
Standard Equates	4,7

This information is stored in the [color_map] section of c5edt.ini, so if you wish you can instead cut and paste Listing 1, replacing your [color_map] section.

Listing 1. Color map settings in c5edt.ini.

```
[color_map]
0=192,192,192
1=0,0,0
2=255,255,255
3=192,192,255
4=192,192,192
5=0,0,0
6=255,255,255
7=255,0,0
8=255,0,255
9=255,255,0
10=0,192,0
11=255,223,223
12=255,192,32
13=0,128,0
14=192,0,192
15=110,110,110
16=0,0,0
17=0,128,128
18=-1
19=-1
20=0,128,0
21=192,192,255
22=192,0,192
23=255,223,192
24=223,255,223
25=0,128,0
26=0,0,192
27=0,0,192
28=0,0,192
29=0,0,192
30=0,0,192
31=0,0,192
```

Figure 2 shows some ABC code displayed with these color assignments.

Figure 2. ABERROR.CLW with modified editor colors.

```

E:\c5\libsrc\laberror.clw
! This is for errors which the program is not expected to Do.
! The default action is to issue a message and HALT
! You may, if you wish, return from this function but it is up to you
! to make sure this 'fatal' condition has been rendered benign
ErrorClass.TakeFatal  PROCEDURE
CODE
    MESSAGE(Self.SubsString() & '  Press OK to end this application', |
            Self.Errors.Title,ICON:Exclamation,Button:OK,BUTTON:OK,0)
    HALT(0,Self.Errors.Title)
    RETURN Level:Fatal

ErrorClass.TakeNotify  PROCEDURE
CODE
    MESSAGE(Self.SubsString(),Self.Errors.Title,ICON:Exclamation,|
            Button:OK,BUTTON:OK,0)

! This is purely here to enable people to add their own fatality levels
! By default an unknown fatality is a program error
ErrorClass.TakeOther  PROCEDURE
CODE
    RETURN SELF.TakeProgram()

! This is for errors which really suggest a mal-functioning program
! The default action is simply to make the error fatal
ErrorClass.TakeProgram  PROCEDURE
CODE
    RETURN SELF.TakeFatal()

```

Of course you should feel free to change these color designations. After all, it's your system! But don't be afraid to spend some time thinking through your color designations. They can be a great asset to you for the time you spend in the editor.

Enjoy!!

Bruce W. Wells is a Systems Analyst Chief with the Kentucky Transportation Cabinet, Division of Information Technology, and is responsible for the design, development and maintenance of applications for transportation issues. He also owns and operates Cardinal Computer Services, a systems and software development and support operation for manufacturing and small businesses. Bruce has been a member of Team Topspeed since 1995.

Copyright © 1999 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.



Clarion magazine

Main Page

Log In

Subscribe

Open Source

Links

Mailing Lists

Advertising

Submissions

Contact Us

Site Index

ClarionMag FAQ

Download PDFs

Search ClarionMag

Subscriber Services

Free User Group Newsgroups

Clarion Magazine provides free newsgroups to Clarion User Groups. These newsgroups are for non-profit activity only. You can use them to provide a chat area, for support, or to organize group projects.

These newsgroups are private to your user group. Clarion Magazine reserves the right to monitor newsgroups and to terminate newsgroups without notice.

Only the contact person needs to be a subscriber to Clarion Magazine. The CUG may distribute the user id and password to all user group members, or to members of other CUGs.

Please allow one week for your application to be processed.

Contact Person

Name

Email address

Contact person's Clarion
Magazine User ID

User Group Information

User Group Name

Location
(City/State/Country)

Newsgroup Information

You can have up to five sub-newsgroups in addition to the top level newsgroup. For instance, if you're applying from the East Overshoe CUG, and your top level newsgroup is called overshoe, you could set up a structure something like this:

overshoe.news
overshoe.projects
overshoe.projects.special
overshoe.support
overshoe.chat

etc
2000
If you're interested
take our poll &
let us know!

Top level newsgroup

(This should be an abbreviation of your user group name).

Newsgroup User ID

(All cug members will use the same user id. This should not be the same as the contact person's user id and it cannot be the same as the user id for the clarion magazine newsgroups.)

Newsgroup Password

Sub-newsgroup #1
(optional)

Sub-newsgroup #2
(optional)

Sub-newsgroup #3
(optional)

Sub-newsgroup #4
(optional)

Sub-newsgroup #5
(optional)

Comments

Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Feature Article

Reporting With Clarion

by Larry Teames

[Last month](#) I explained what "areas" and "bands" are in the context of Clarion for Windows reporting. This month I'll begin looking at the Report Formatter available in the Clarion5 IDE.

However, rather than reiterating all the same information already available on this subject in the Clarion5 Help file, I'll try to limit my coverage to tips, tricks, and other tidbits of information that I think may help you complete your reporting tasks faster and with less effort.

For the record...

The Clarion Report Formatter can be very intimidating and frustrating when first used if you don't have a clear understanding of how it can be used. In effect, the report formatter is multiple formatters, each of which is almost identical to the window formatter. All of these formatters are linked together in a single interface. Of course, there are some control types that do not have any practical use in a report, and those types are not available in the formatter.

Tip: Save yourself some grief and forget about using the listbox control that's available in the formatter. It can seldom be made to produce desired results, and TopSpeed has gone on record stating that it will be removed from the formatter sometime in the future anyway.

From a user's perspective, each band is handled independently by it's own (window) formatter. And since they are all accessible at the same time, you not only have the ability to see how the controls align and relate to one another across bands, but can also perform some functions like control alignment against multiple controls in multiple bands in a single operation.

Before I go any further though, there are two very important rules that you should follow when formatting reports:

1. Save your report and application frequently!
2. Save your report and application frequently!

The report formatter does mostly good things, but occasionally it does some not-so-good things. So, save your work frequently by getting completely out of the formatter, and saving the application. I also tend to reboot my machine about every hour or so when doing intensive report formatting and testing.

Tip: I run the Resource Meter that comes with Windows 98 (RSRCMTR.EXE) from my StartUp group, and when the resources get to 40% or so, I always reboot

Rebooting gives Clarion (and your machine) a chance to recover some of the resources that may get locked or lost by the formatter, or other elements of the IDE.



If you're interested
take our poll &
let us know!



I have created several thousand reports since I started using Clarion, and I simply don't have most of the weird problems that many users report related to using the report formatter. I believe that this frequent saving/rebooting is what makes the difference.

Before you proceed, you should read the following topics in the Clarion5 Help file in their entirety: "How to Use the Report Formatter - An Overview" through "How to Print Labels" (just keep hitting the >> button till you've gone through all the topics).

Doing so will assure that you are familiar with at least the basics of how the formatter is used. Be sure to pay particular attention to the topic "How to Control Page Breaks", since this discussion explains a great deal about how the Report Engine works. Of course, if you have already used the formatter to create numerous reports and feel that you already know the information contained there, you can skip the reading assignment.

Get Ready, Get Set...

What's the best or most correct way to design a report? In my opinion, there is no single best or correct way to design most reports. I'm certain that what works best for me is probably not the best way for everyone. However, in this column I will approach things in my own way, and you can try what I recommend, or not, as you like. The important thing is to try and understand the concepts and ideas that I describe and then use them in a way that complements your own work style.

I expect that most of you have already created several reports, and have experienced some level of frustration in doing so. One of the main reasons for these frustrations is that a great deal of what gets done on your reports is automatic. Although this can be a good thing, it can also be an irritation. I have found that in many of these situations, the only way to get the most from my efforts is to completely bypass the automatic functionality and write code to perform the same functionality and/or extend it to do that extra bit of work that the built-in functionality doesn't offer.

With all that in mind, let's take a look at the report formatter.

Exploring the Report Formatter

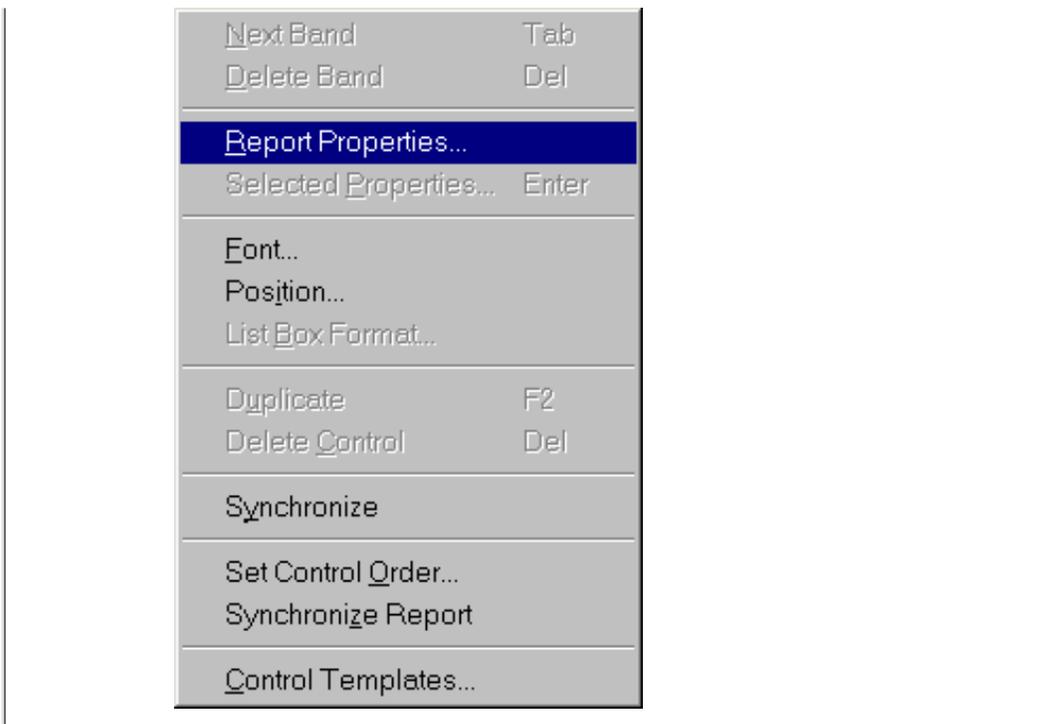
Figure 1 shows the main menu of the report formatter (RF):

Figure 1. The report formatter main menu.



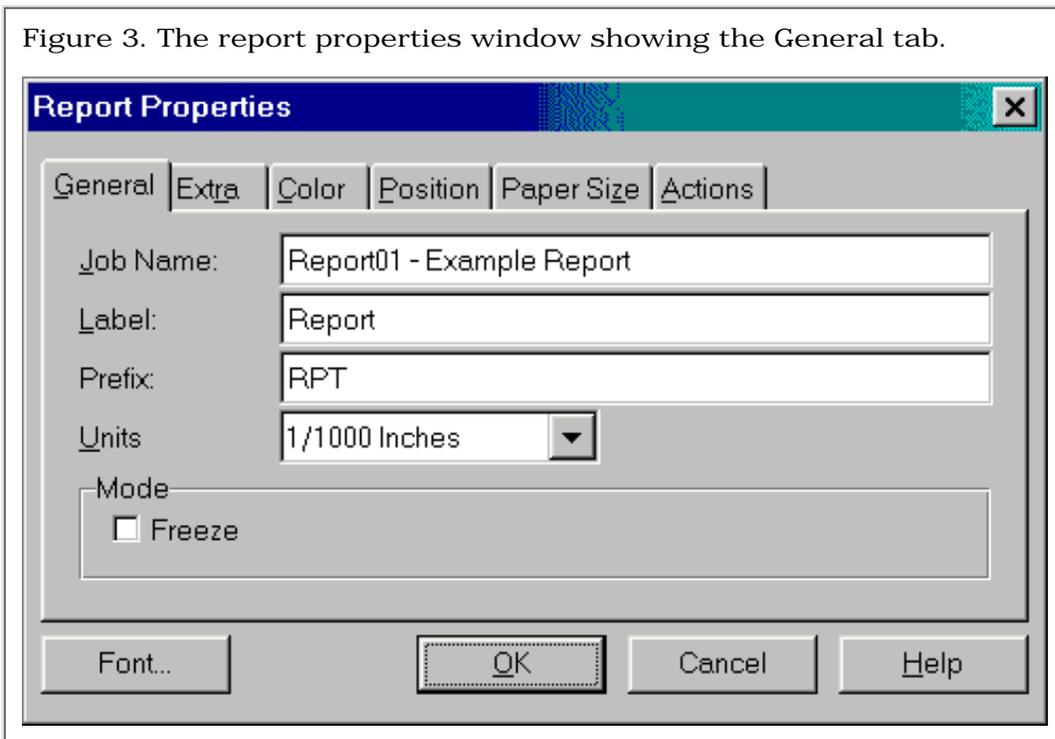
On the EDIT menu (see Figure 2) probably the most frequently used selection is Report Properties, since it allows you access to most of the attributes of the report format.

Figure 2. The report formatter Edit menu.



Selecting the Report Properties menu option displays the window shown in Figure 3.

Figure 3. The report properties window showing the General tab.



The contents of the Job Name field will appear as the Job Name of the Windows Print Spooler. In a newly generated report, Job Name will be blank. I'd suggest that you always enter something into this field of every report, so that you (and your customers) can easily identify the report when it's spooled under Windows.

TIP: you can set/change this field at runtime by setting `REPORT{PROP:TEXT}='My Report Name'`. Do this in the After Opening Report embed, so that the change takes place before any print lines are generated.

Although you can change the Label and Prefix of the report, I always leave them set to their defaults ("REPORT" and "RPT"), since this is about as intuitive as it can get, and if all reports are referred to by the same name and prefix I don't have to remember differences from procedure to procedure.

Generally, I leave UNITS set to thousands of an inch, because this provides the finest resolution (for positioning controls, etc.) that most printer drivers are capable of. If you have a particular report which is best handled with millimeters, as may be the case when a preprinted form is formatted in MMs, then feel free to switch.

Figure 4. The report properties window showing the Extra tab.

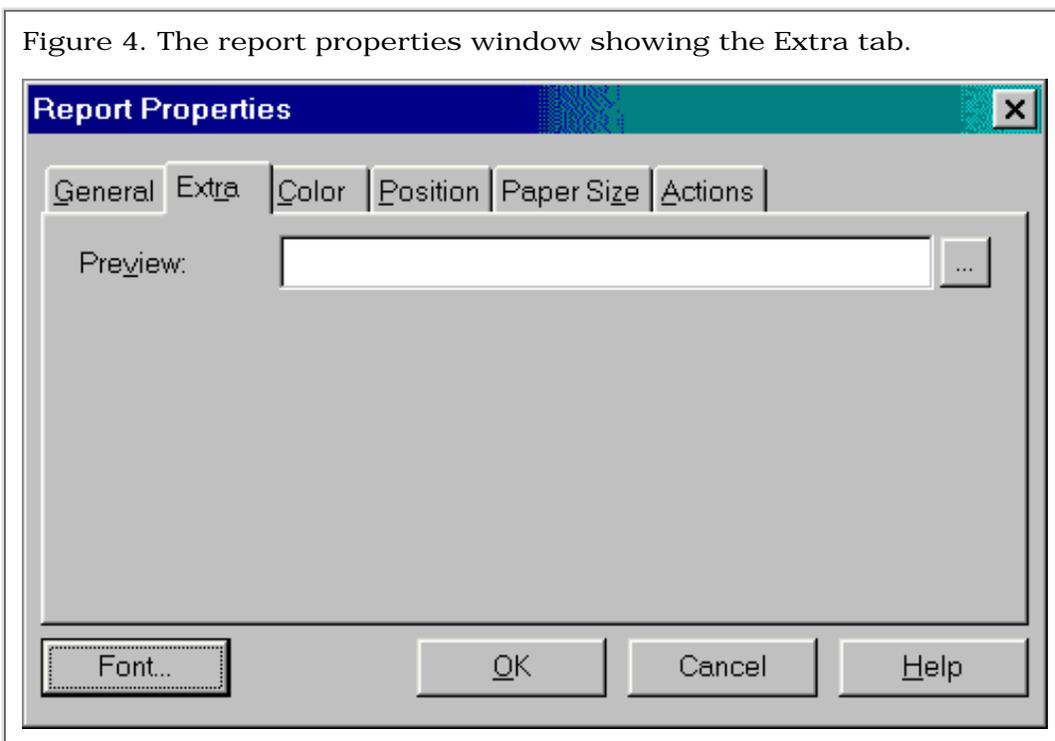
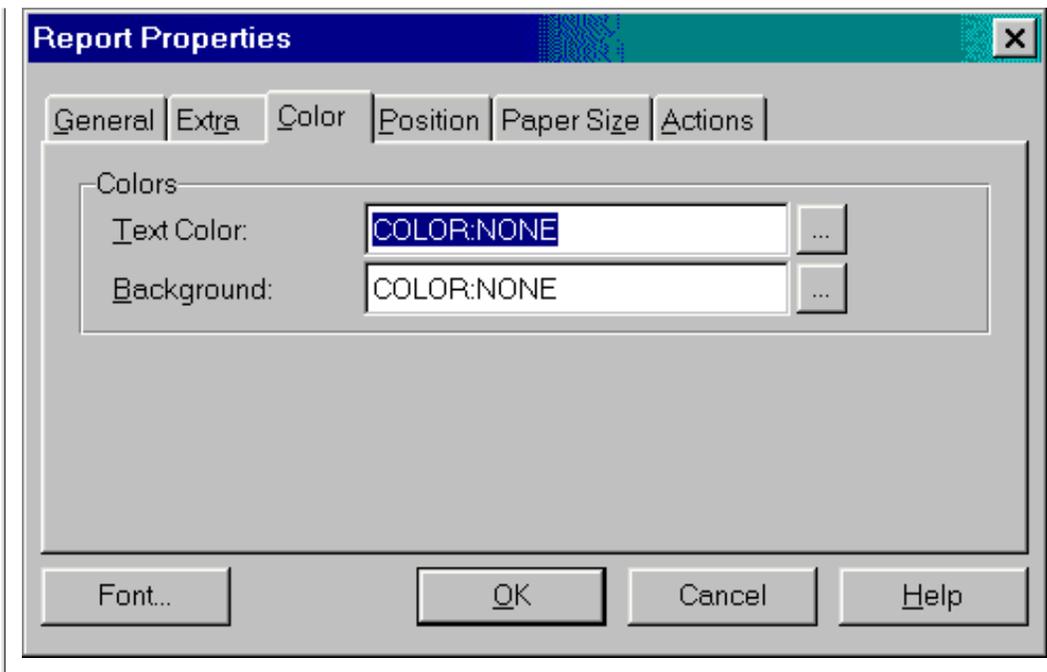


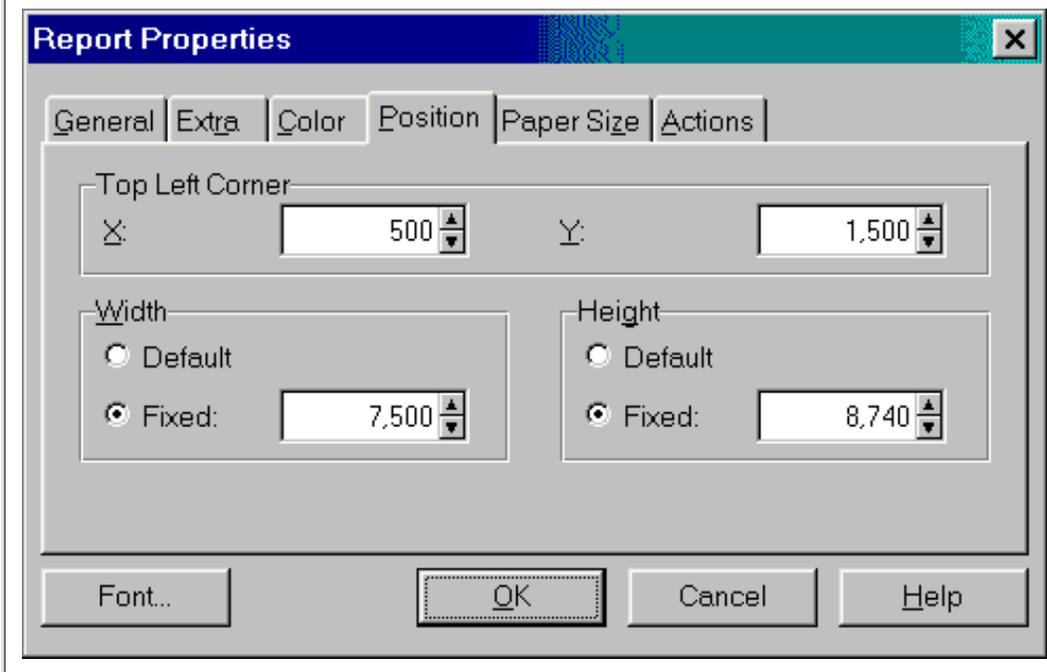
Figure 4 shows the Preview field on the Extra tab. As the help file states, **LEAVE THIS FIELD BLANK!** Failure to do so tends to end with a GPF at runtime.

Figure 5. The report properties window showing the Color tab.



The Color tab (Figure 5) allows you to override the default Text Color and Text Background used on the report. My experience is that you will be better off by leaving these set to the default.

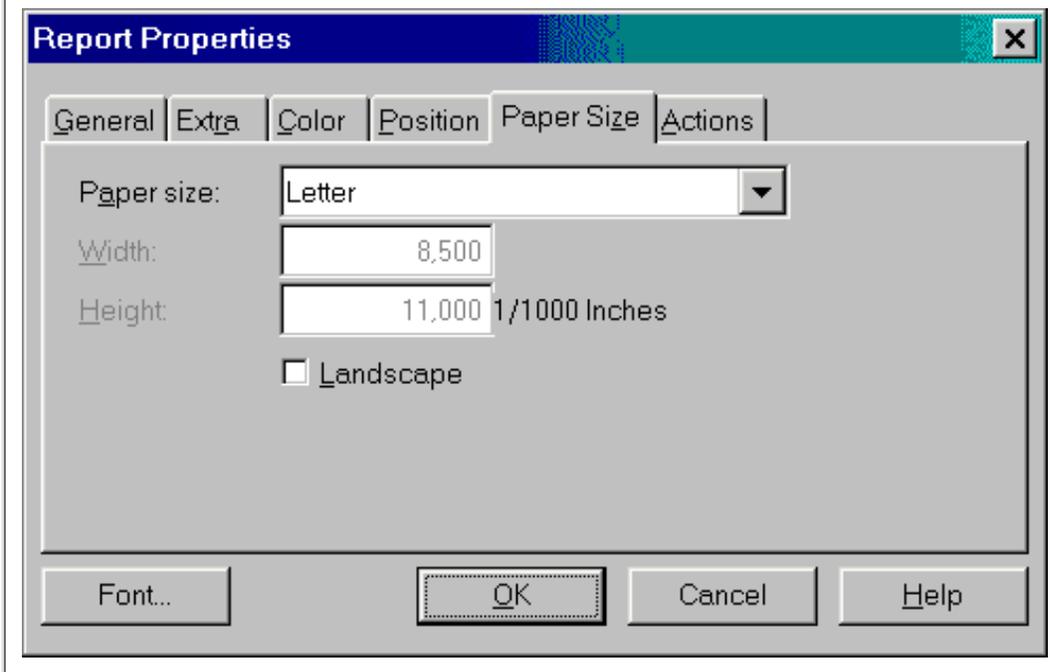
Figure 6. The report properties window showing the Position tab.



The Position tab (Figure 6) provides you with a way to set the position and size of the Detail Area (remember last month's subject?) of your report. Remember, you can also do this interactively by using the Page Layout View menu option ([also described last month](#)).

The X and Y positions define how far from the left and top edges of the paper you want the Detail Area positioned. Unless you just enjoy trying to figure out why your report acts so weird on different computers, never set the height or width to default.

Figure 7. The report properties window showing the Paper Size tab.



The Paper Size tab (Figure 6) is where you define the dimensions of the actual piece of paper that your report will print on. In addition to the numerous predefined paper sizes that are available from the drop down list (Letter, Legal, A4, etc.), you can also specify a page size other than those listed by selecting "Other" from the drop down list, and then entering the actual width and height of the paper your report will print on.

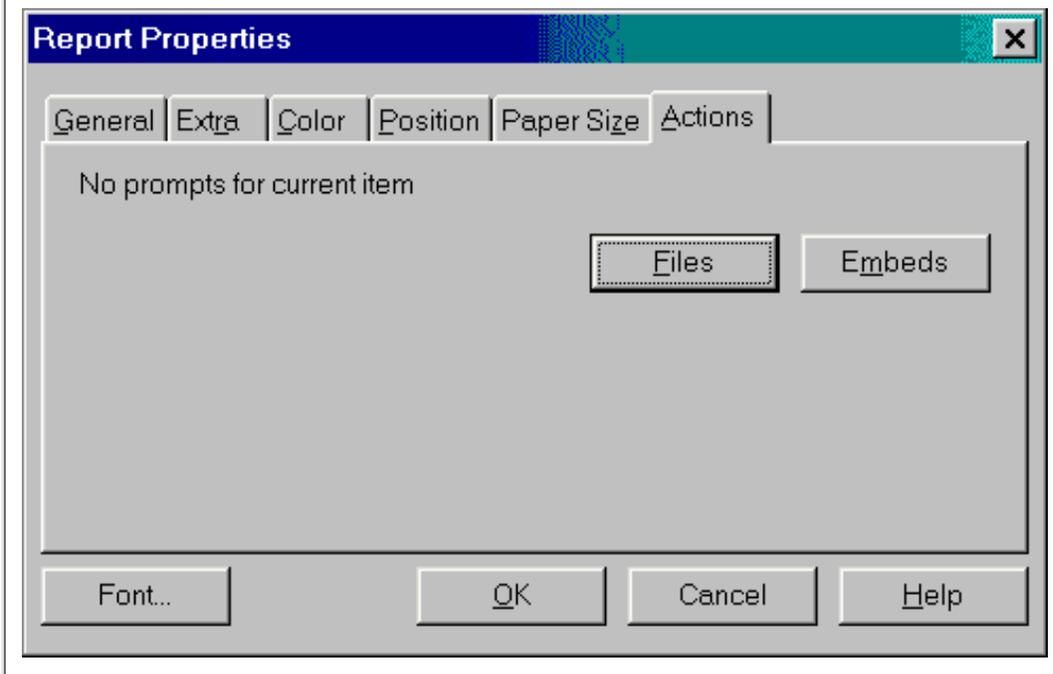
You should assure that the printers and printer driver(s) that your customers have will support custom paper sizes before you design a report using a non-standard size. You can do this by viewing the Properties of the printer driver in question, selecting the Paper tab, selecting the Custom paper size, and entering the dimensions of the page you want to use. If the driver does not allow you to perform this operation it will not support the paper size you are setting the report to.

TIP: to perform this test on a printer that you don't have attached to your computer, simply install the driver for that specific printer, then perform the test. When done, you can delete that printer driver.

Note that nearly all laser printers, and compatibles like DeskJets, are designed to use only a few fixed size papers, so defining other sizes will not be possible for these types of printers.

If you do try to use an unsupported paper size with any printer driver, the Clarion Print Engine will simply set the paper size to the default paper specified by the printer driver, and continue on with the report.

Figure 8. The report properties window showing the Actions tab.



The Action tab (Figure 8) has buttons for accessing the file tree, and the embed tree of the report procedure. My personal work habits have never led me to use these buttons since they don't seem to provide benefit over saving the report format, and exiting the formatter to access the file and/or embed tree. I suspect that continual use of either of these buttons during a formatting session would probably contribute to the probability of losing your work, but that's pure conjecture on my part.

The Font button on the Report Properties window allows you to specify the default font to be used when formatting and printing this report. The default font is the font used on all controls where no specific font is indicated. I generally prefer to use a TrueType font like Arial for most of my reporting. The TrueType fonts render better for previewing than do fixed pitch fonts.

The rest of the main menu choices are self explanatory or described well enough in the help file that they don't really need more description by me at this time.

What's coming up ...

Over the next few months, I will be designing a number of reports to illustrate approaches that can be used to handle various reporting issues, both common and uncommon. To make it easier for you to follow and later review what I cover, I will provide for download the APP, DCT, and data files used in these examples.

Hopefully as this series proceeds you'll begin to feel less frustrated and more in control when designing your reports.

Till next month, Happy Reporting!

Larry Teames is an independent software developer, and one of the four founding members of Team TopSpeed. He is also president of [Creative PC Solutions, Inc.](#) which markets the popular Clarion 3rd party product Creative Reporting Tools.

Copyright © 1999 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.



Clarion magazine

[Main Page](#)
[Log In](#)
[Subscribe](#)
[Open Source](#)
[Links](#)
[Mailing Lists](#)
[Advertising](#)
[Submissions](#)
[Contact Us](#)
[Site Index](#)
[ClarionMag FAQ](#)
[Download PDFs](#)
[Search ClarionMag](#)

Feature Article

Sorting On The Run (Part II)

Creating Tabs and Adding New Sort Orders:

by Steve Parker

In the [first installment](#) of this series I explained that when browse tabs are created, information is taken from the prompts on the Browse Box Behavior ... Default Behavior and Conditional Behavior options. These include locator, key, filter and Additional Sort Fields information.

Information is collected starting (logically) with the second tab. The default locator, filter, key and Additional Sort Fields are collected last. In this way, options on the Default Behavior tab become the target of an ELSE clause (i.e., the default action).

These data are added to a private queue `BRWx.SORT` in the browse object. When the end user changes tabs, the tab number (`Choice(?CurrentTab)`, actually) is used to calculate a pointer to retrieve the information from the queue.

The `AddSortOrder` method actually adds the information to the queue. `AppendOrder` can add additional information to the sort specified in `AddSortOrder`. And, most importantly, `AppendOrder` can take a variable as its parameter.

But First ...

Before attempting to implement additional orders there is a small housekeeping issue that must be resolved: dynamically creating the required number of tabs. After that, the remaining tasks are:

1. add additional sort orders
- and
2. correctly associate created tabs with their target data,

which will complete the implementation of sorts on the run.

Creating Tabs

If, like me, you are not familiar or comfortable with creating controls dynamically, now would be a good time to briefly check out "Create (create new control)" in the on-line help, before proceeding.

The code to create controls should be placed after the window is opened so that the structure required to receive the controls (the window) is already created in memory. But new controls must also be created before the window is displayed. (Technically, you can create a control any time you want. Because it is hidden, you will need to both unhide it and refresh the window. Considering what I'm trying to accomplish, however, after opening the window is the appropriate place to create the new controls.) In this respect, creating a control is no different than modifying a control when opening a window. This implies that the

etc
2000

If you're interested
take our poll &
let us know!

BKO
Enterprises, Inc.

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

ThisWindow ... Init ... Open the window embed is where the tab creation code should be placed. Creating tabs is a bit different than creating controls used for data capture.

First, unless you need to address the new tab directly in code, for example

```
SELECT(?MyNewTab)
```

a Use variable, the first parameter of Create, is not necessary. Nor is it necessary to set {Prop:Use}.

Second, because a tab must be on a sheet, the Create statement's parent (third) parameter is required.

Generically, the code to create a tab looks like:

```
MyNewTab = CREATE(0,Create:Tab,?CurrentTab)
```

Note the zero in the first parameter. The documentation states that this parameter is optional. However, as sometimes happens, the compiler does not agree. When the documentation and the compiler disagree (and until that disagreement is resolved), it is wise to go with the compiler. Therefore, supply a dummy value for the first parameter, the "field number or field equate."

By default the tab has no text, so the next line of code should assign some text:

```
MyNewTab{Prop:Text} = 'Hey look me over!'
```

And, per the docs, should you want the end user to actually see the tab:

```
UNHIDE(MyNewTab)
```

(This last step is not, strictly speaking, necessary. But why go to the effort otherwise?)

It's time to apply this knowledge and create the tabs.

The easy way to create exactly the right number of new tabs and to do so in one go is to put the CREATE statement in a loop.

Suppose the file containing the user defined sorts is UserSort, its prefix is USR and it contains a field called Description. Then the code in Listing 1 should do the job.

Listing 1. Creating tabs.

```
i = 0
NDX = 0
ACCESS:UserSort.UseFile()           !Ensure file is opened
SET(UserSort)
LOOP NDX = 1 TO RECORDS(UserSort)    !Only as many as the user
                                     !has actually "requested"
    NEXT(UserSort)
    i = CREATE(0,Create:Tab,?CurrentTab)
    i{Prop:Text} = CLIP(USR:Description)
    UNHIDE(i)
END
```

If you are thinking that a safety check, like:

```
If ACCESS:UserSort.Next()
  Break
End
```

should be placed after the `Next()` statement and you routinely do this sort of check, the code will fail to work as expected. If there is only one user defined sort in the file, one record, reading it will trigger the `Access:UserSort.Next()` condition and the tab will not be created.

If you are uncomfortable without an error check, try placing it after the `UNHIDE()` statement.

If you stop at this point and make your app, the tabs will be created just as expected. They won't do anything, of course, but they will be there and they will be there in record order. (For this reason, in a real application, you would probably not `SET()` the file but a key.)

So far, so good.

Note: if you also intend to use a toolbar, your variable must be at least a `SHORT`. Toolbar buttons use field equates from 2000-13. The `CREATE` statement returns the field equate, so the first `CREATE` with a toolbar in place will return 2014. The variable must be long enough to receive a value this large. A `LONG` would be better since that's the return value of `CREATE`. (Tip: It sometimes helps to read the documentation all the way to the end, not simply stopping at the point where you understand. This is the voice of experience speaking.)

The greatest concern with these created tabs must be the accuracy of `Choice(?CurrentTab)` when a runtime tab is selected. If you, wearing your developer hat, cannot be absolutely certain of this value, there will be absolutely no way to set the desired order.

Without sure and certain knowledge of the value of `CHOICE(?CurrentTab)` (or a way of calculating it), you have no way to calculate the requested order, retrieving the correct queue entry.

Remember, when a new tab is selected, `CHOICE(?CurrentTab)` is used to calculate a pointer into the queue containing the sort information, similar to Listing 2.

Listing 2. Setting sort orders by tab choices.

```
IF CHOICE(?CurrentTab) = 2
  RETURN SELF.SetSort(1,Force)
ELSIF CHOICE(?CurrentTab) = 3
  RETURN SELF.SetSort(2,Force)
ELSIF CHOICE(?CurrentTab) = 4
  RETURN SELF.SetSort(3,Force)
ELSE
  RETURN SELF.SetSort(4,Force)
END
```

This is an issue of some import. You need the queue entry number and `Choice(?CurrentTab)` is the only publicly available datum related to it.

However, it turns out that this is not a problem. Because the new tabs are created after the tabs created during window design, the new ones will automatically attach to the Sheet after those tabs. That is, if you create three tabs in the window painter, the first dynamically created tab will, in fact, be in position four.

You may easily satisfy yourself in this matter by embedding:

```
MESSAGE( CHOICE(?CurrentTab) )
```

in the `?CurrentTab ... NewSelection` embed. You will always get the return value you expect.

So all is well here.

Adding New Sort Orders

Ok, you now know how to create tabs in the desired order and you know, in principle, how to add sort orders. It's time to add some actual sort orders.

Two things should be obvious.

First, supposing the file variable you want to use is labeled `USR:Order`, you can add `USR:Order` to the queue with:

```
BRW1.AddSortOrder(, )
BRW1.AppendOrder(USR:Order)
```

(Note the absence of apostrophes, indicating that the use of a variable.)

Second, you need to add the new order(s) after the template generated orders.

If new orders are added before the generated key sorts, the calculation to retrieve information from the queue will be hopelessly mashed. The added orders will start at entry number one but the templates will retrieve those entries when tabs two, etc., are selected.

The first available embeds after the generated `AddSortOrders` are in the `Resizer` method (see Figures 1 and 2).

Figure 1. Embed points for adding sort orders.

```
BRW1.AppendOrder(' +Cus:LastName' )
BRW1.AddSortOrder(, )
BRW1::Sort0:StepClass.Init(+ScrollSort:AllowAlpha)
BRW1.AddSortOrder(BRW1::Sort0:StepClass,CUS:CustomerNumber)
BRW1.AddField(CUS:Number,BRW1.Q.CUS:Number)
BRW1.AddField(CUS:LastName,BRW1.Q.CUS:LastName)
BRW1.AddField(CUS:FirstName,BRW1.Q.CUS:FirstName)
BRW1.AddField(CUS:CompanyName,BRW1.Q.CUS:CompanyName)
? Initialize resizer
Resizer.Init(AppStrategy:Surface,Resize:SetMinSize)
SELF.AddItem(Resizer)
? [Priority 8140]

Resizer.AutoTransparent=True
? [Priority 8160]

? Start of "After initializing resizer"
? [Priority 5000]
```

Adding entries to the sort order queue in the Resizer! Don't let this make you crazy. This is just a place in the code, no more. The important thing is that nothing affecting the window, sheet or any existing tabs has occurred at this point.

Listing 3. Creating the sort orders.

```
i = 0
SET(USR:SortOrderKey)
LOOP i = 1 TO RECORDS(UserSort)
  NEXT(UserSort)
  BRW1.AddSortOrder( , )
  BRW1.AppendOrder(USR:Order)
END
```

Unfortunately, because the `Sort` property and, specifically, the queue referred to in the documentation is private, the only way to be certain this code is good code is if it works. Unfortunately, the problem is that if the browse does not behave as expected you cannot immediately conclude that there is something wrong with the solution. This is the stuff of which bugs are made. The problem may be in the execution of the solution.

Unfortunately, I can tell you with some confidence that the solution described here will work; it adds the expected data to the queue. But, it does not work quite as expected (a sample `.APP`, as well as a finished program are available for download; if you run the app, you will see that the "Single Tab - Uncorrected" menu option will behave as described below).

If you create both embeds discussed here and test the app – without yet creating the code to apply new sort orders – your browse will behave bizarrely. When the browse is opened, it will be sorted by the last order added. It will not be sorted by the default key, if any, or first order added.

Nothing in the documentation suggests this should happen. But it is obvious that `AddSortOrder/AppendOrder` not only adds information to the `Sort` queue, it also applies it. Thus, the last added order is active – instead of the expected order – when the browse is first opened.

A fix-up is equally obvious.

It's clear that the `SetSort` method sets the active order and that the browse object is `BRWx`. You know exactly how many tabs you placed on the window in the window painter.

Therefore, where `n` is the number of tabs on the window before adding custom tabs, the queue entry corresponding to the default sort is also `n`.

Using the code in Listing 2 as a guide, queue entry `n` is actually the default order for the browse.

To reset the browse, just add:

```
BRW1.SetSort(n,1)
```

immediately after adding all the supplemental sorts.

The final code for adding the sort orders, then, looks like Listing 4.

Listing 4. Final code for adding sort orders.

```
i = 0
SET(USR:SortOrderKey)
LOOP i = 1 TO RECORDS(UserSort)
  NEXT(UserSort)
  BRW1.AddSortOrder( , )
  BRW1.AppendOrder(USR:Order)
END
BRW1.SetSort( n , 1 )
```

Unfortunately (aren't you getting tired of that word?), I'm also out of space for this installment.

Next Time

Next time, I'll finish the application by adding code to associate the supplemental sort orders with the correct tab. This will ensure that the desired sort will activate on a tab change.

In the meantime, [download the sample .APP](#) to see how adding tabs and orders is accomplished.

[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 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.



Clarion magazine

[Main Page](#)
[Log In](#)
[Subscribe](#)
[Open Source](#)
[Links](#)
[Mailing Lists](#)
[Advertising](#)
[Submissions](#)
[Contact Us](#)
[Site Index](#)
[ClarionMag FAQ](#)
[Download PDFs](#)
[Search ClarionMag](#)

Feature Article

Interview: Ragnar Hellspong

In the competitive world of shareware many products appear, but few are successful. One of these is Ragnar Hellspong's [ForKeeps](#) program, which archives mail, CompuServe forum, and newsgroup messages. ForKeeps is an award-winning product written in Clarion and used worldwide (and by not a few Clarion developers). Visit the [ForKeeps web site](#) for a [30 day trial version](#).

Clarion Magazine: Tell me a little about your background.

Ragnar Hellspong: I'm located in Stockholm, Sweden and I guess my background is pretty unusual for a Clarion developer. I have a long career in the electronic and electro-technical industry behind me, starting out as a sales engineer in 1969 and ending as general manager in 1990. At that time I left the industry to develop my own sales automation system, a decision more based on my knowledge and ideas about sales and marketing automation than on software development skills. In fact I had very little software skills at the time.

How did you get started with Clarion?

I started out with a development system called FOCUS from Information Builders, without realizing the limitations of that particular system. After about 6 months I saw an ad about Clarion 2.1 and liked what I saw, especially the part about royalty free runtime licences. Since then I have been using nothing but Clarion - until recently when I was forced by a customer to make an application in Visual Basic 6.

What gave you the idea to create ForKeeps?

I joined the Clarion forum on CompuServe early in 1994 and used WinCIM to collect my messages. Where I live there are no other Clarion developers and the Clarion forum was a real blessing for me as I could suddenly communicate and share information with lots of other Clarion developers.

During 1995 and early 1996 the traffic volume increased very much on the forum and I was more and more frustrated that I wasn't able to find the information I needed in WinCIM. To keep it working I had to zip up messages older than a month and once zipped up the information wasn't available anymore. Thus I got the idea that I should read the messages into a database where I could keep the entire message base available for search.

When did the first version go into beta, and how did you find beta testers?

That was in June 1996 and I used Jeff Slarve as a beta tester. He mentioned on the Topspeed forum that he was looking for a forum message archiver like the one I was working on and I said I had something going and asked him to Beta test it for me. Wes Gates did some testing also. Since version 1.0 I have released 20 (!) upgrades and a lot of other people have been involved in beta testing over time.

If you're interested
take our poll &
let us know!

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

ForKeeps 1.0, which was finally released on September 1, 1996, was a forum message archiver only. The mail part was added with version 2.0 in December 1996 and the newsgroup part with version 3.0 in October 1997.

What was the biggest challenge in getting that first release out the door?

I had very few problems actually. As I was hand coding I wasn't affected by the various template bugs that people complained about. One design problem I remember was that my home grown keyword indexing was rather slow. This problem wasn't solved until a year later when I could replace it with Jim Kane's WordSearch engine.

What made you decide hand-coding was a better approach than using the templates?

It was not a decision especially for ForKeeps. I have always hand coded - from CPD 2.1 and on through Clarion 5. I don't think hand coding is a better approach than using templates but it works for me. It also allows me to optimize my code and keep it as tight as possible. Keeping the size as small as possible is important for a shareware program that must be downloaded.

What kind of response have you had from users?

A very positive and sometimes enthusiastic response. It seems that ForKeeps is an important piece of software for many users.

Are users ever curious about how ForKeeps was created, i.e. what language was used?

Very few people ask that question. On the other hand it is mentioned on my web site and in the Help file so maybe they don't have to ask. Before I added the export facility some people asked me about the database file format and were concerned about their messages being locked in. That isn't an issue anymore.

ForKeeps has a modular structure. Would you recommend this approach to other developers?

I started using the DLL approach already with CFD 3 and I have never made stand alone EXEs since. In ForKeeps I have a thread manager DLL (based on the Harms-Santos book TM code by the way), a PowerBrowse DLL and a general DLL, containing the import part and most of the generic functions I need.

The thread manager I haven't touched since it was made in 1996. The PowerBrowse DLL I had to recompile with a new PB library from CW2 to CW4, but otherwise it's untouched also. The big advantage with the DLL approach is that I can use the generic DLLs in all my applications. That's especially important for me as I am still hand coding.

I understand ForKeeps has received awards and other recognition. Can you provide some details?

ForKeeps was selected WUGNET's Pick of the Week as an exceptional shareware program on the Windows Utility forum in October 1996 and on the Windows Users forum in December 1996. In January 1998 ForKeeps was given a top Excellent shareware rating by PC World's French edition INFOPC. The latest version 3.93 was reviewed recently by ZDNet Software Libs and earned a top five star rating.

How did Jim Kane come into the picture?

He became an early user in September 1996 and later we exchanged some ideas about searching algorithms. Then in July 1997 he developed his WordSearch engine and also some sort of WordSearch add on for ForKeeps that he sent me for evaluation. I saw the possibilities for a much improved search function in ForKeeps and wanted to integrate it completely rather than sell it as an add on. So with version 3.0 it was an integral part of ForKeeps, sold as an extension for the professional user.

What's the ratio of downloads to registrations? Do quite a few

people register after the 30 days are up or do they buy straightaway?

I have no idea about the download rates anymore. In the beginning I was eagerly following them on a weekly basis but today ForKeeps can be downloaded from so many sites on CompuServe and the Internet that I cannot keep track anymore.

Shareware packages have a way of propagating from one shareware site to another by themselves and some shareware sites - for example Simtel - are mirrored all over the world. I have also found ForKeeps included on PC magazine cover CDs with and without my consent.

Few people seem to buy before the 30 day test limit times out, however I got the first order on version 1.0 one day before I uploaded the program so there are exceptions. Jack Hobbs was the brave buyer.

What has been your most effective marketing technique?

I haven't really been marketing the program. No ads, no mass mailing, no press releases. I have made it available for download and free test and I am supporting registered users and potential users by mail, on a ForKeeps section of the CompuServe Freeware forum and on a ForKeeps newsgroup, that's all. I'm also very responsive to user problems and I feel that users appreciate that a lot.

In other words - ForKeeps is sold by word of mouth.

How is Clarion suited or not suited to developing commercial applications like ForKeeps?

It's ideal. As an example ForKeeps is self contained in its program directory. No fussing with the registry or AUTOEXEC.BAT. In contrast most Visual Basic applications are very dependant on a proper set up by the user and lot's of things can go wrong.

For custom-tailored applications the customer wants to approve the development tool, which is bad for Clarion because it is not well known. For commercial standard applications the development tool is completely irrelevant to the end user.

Do you have any long-range plans for ForKeeps you'd like to share with ClarionMag readers?

I'm working now on the next major version 4.0. All enhancements will be concentrated on the mail and newsgroup parts. In the mail part Outlook 98 and the new Outlook Express 5 format must be supported.

In the newsgroup part I will adapt to the strong trend from public Usenet newsgroups to private newsgroups. Two years ago there was but one Clarion newsgroup. Now there are plenty, much like sections on the CompuServe forums. This makes global searching a real problem. I will therefore allow redirection so that one ForKeeps database can contain any number of redirected newsgroups as sections. That will simplify word indexing and searching a lot and allow me to filter out cross postings.

In the future I must probably deal with the trend towards HTML and RTF format messages. Today they are simply converted to plain text but sooner or later I may need to handle them in their native format.

The need for an archiver like ForKeeps can only grow. There are no direct competitors in sight although indirect ones like DejaNews for Usenet search and various knowledge bases for specific areas are available as substitutes. For mail there is no competition whatsoever.

If you have a Clarion success story you think Clarion Magazine readers should know about, send an email to editor@clarionmag.com.

Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Categorized Clarion Sites

Updated May 17, 1999

This page contains listings for Clarion-related web sites, loosely categorized into the headings listed below. Many sites appear under more than one heading. If you have a URL you'd like Clarion programmers to know about, send it to links@clarionmag.com.

Index

- [Data Entry](#)
- [Graphics](#)
- [Knowledge Bases](#)
- [Reports](#)
- [Software Development Tools](#)
- [Security](#)
- [SQL and ODBC](#)
- [Miscellaneous](#)
- [Clarion Information](#)
- [Consultants](#)
- [Web Design](#)
- [Internet-Related](#)
- [Linux](#)
- Distributors

*You don't
want to
miss even
one issue*

subscribe to
Clarion
magazine

CLICK HERE

Data Entry-Related Products

[NextAge Consulting](#)

NextAge Consulting provides a full range of consulting services to our clients. This includes software development, website design and hosting, IT and project management, and joint ventures with other software developers. Most of our team of employees and subcontractors have been providing consulting services for 10 to 20 years in a wide range of industries and technologies. "Feedbackware" templates include application images, multicategory lookup, security, and miscellaneous. Posted on February 15, 1999

[Gitano Software](#)

RenTrack property management software, various third party tools including calendars, calculators, notes management, print

options/preview,
Posted on February 15, 1999

[Logic Central](#)

Mike Pasley's site - "Logic Central is my web site where I'll have application engineering items that I find interesting and hope you find interesting as well. Of course, there is a focus on Clarion from Topspeed Corporation. I run across things that helps me better use the Clarion development environment and I would like to share some of these items. So, I hope that this site presents to you useful concepts, old reminders, or new ideas that challenge you or stimulates discussion with others."
Source for "Simplates"
Posted on February 15, 1999

[ProDomus Inc.](#)

CWIntl -- International Localization, PD Translator International Localization, PDAddress -- Worldwide Address Formats, PD Lookup -- Better Lookups: PDLookup, browse button lookup. PD Drops drop lookups. PD Date Tools - Scrolling date objects, date functions, time zone information, and threaded calendar. Localized date function rules, month names and day names. PDFin - Finance Toolkit Simple Tools -- Article and source code example about developing 16 and 32-bit libraries. Includes modified mangle program and source code (prj, exp, clw, inc, and tpl files)
Posted on February 15, 1999

[Software By Ragazzi](#)

"We provide quality add-ons which will give your Clarion programs a more professional look while making the job of programming easier and faster. Our line of templates and libraries allow you to concentrate on developing an application specific to your needs and leaving the mundane aspects of programming to our product line."
Posted on February 15, 1999

[J&S Software Co.](#)

" We have been writing custom business software since 1991. Our programming language of choice is Clarion for Windows. It is the finest business database language around. Along with the custom software development, we also write add-on tools for other Clarion for Windows (aka CW) developers. We are very proud of our Mailmerge add-on. Download the demo and check it out!" Producers of In-Back.
Posted on February 15, 1999

[NT Tools & Clarion Tips/Tricks](#)

Clarion 5 Tips and Tricks. Preview the latest MS-SQL Tips and Tricks for C5. API Call to Get the OS version. Hot-Keyed fields and Entry Color fields FEB99.ZIP. Handout. Sample objects are used. API call to capture the Win NT / 98 / 95 OS User Name.
Posted on February 15, 1999

[Tinman Development Corp.](#)

DAS Registration, DAS Tools, DAS Security, DAS Quickhelp, Tintools (free), DAS tagging (free)
Posted on February 15, 1999

[Locus Software](#)

You can download free templates for Clarion 2.00x and Clarion C4. The templates for C4 works with the normal Clarion and Clarion ABC template chain. The templates contain: Inplace edit of browse-box with validation and lookup. File manager for updating files after changes to the dictionary. Rich Edit Control to view and edit *.rtf files and memo-fields with Mail Merge. Query by Example view and report of all files in your application Report filter with an improved report generator. Popup toolbar, popup date and popup directory. Popup help generation. Page tree. And more.....
Posted on February 19, 1999

[BoxSoft Development Inc.](#)

"Mike Hanson's Super Templates for Windows expands Clarion's Application Generation power. You'll develop higher quality programs that your clients will pay a premium for. Mike's Models and Templates for

DOS have been popular for years, and now are available for Windows. Get the features you need. Mix and match these templates using Clarion's template registry." Many, many features, some free templates.
Posted on February 23, 1999

[Mitten Software](#)

"Mitten Software provides time-saving tools and services for Clarion Developers. We support all versions of Clarion Software, both Windows and DOS. Contact us for a wide variety of programming tools, templates, books and educational material. Ask for help with your Clarion Software projects. We provide project management, consulting and contract programming."

Posted on February 23, 1999

[Berthume Software and Consulting](#)

Addons, Electronic Forms Solutions, and Custom Software Development. Products include Powerrun to launch processes in foreground or background, PTools for printing reports from disk, PCL Tools, which lets you use PCL commands in a report, and Transform Professional which creates forms from scanned images.

Posted on March 3, 1999

[Princen Information Technology](#)

Templates to directly send internet mail, control user prompts, play multimedia files, print dictionary, show directory listbox, limit threads, print utilities, DDE to control Word/Excel.

Posted on March 27, 1999

Graphics

[NextAge Consulting](#)

NextAge Consulting provides a full range of consulting services to our clients. This includes software development, website design and hosting, IT and project management, and joint ventures with other software developers. Most of our team of employees and subcontractors have been providing consulting services for 10 to 20 years in a wide range of industries and technologies. "Feedbackware" templates include application images, multicategory lookup, security, and miscellaneous.

Posted on February 15, 1999

[Gitano Software](#)

RenTrack property management software, various third party tools including calendars, calculators, notes management, print options/preview,

Posted on February 15, 1999

[Software By Ragazzi](#)

"We provide quality add-ons which will give your Clarion programs a more professional look while making the job of programming easier and faster. Our line of templates and libraries allow you to concentrate on developing an application specific to your needs and leaving the mundane aspects of programming to our product line."

Posted on February 15, 1999

[CW Addons \(Lodestar Software\)](#)

"Lodestar Software is proud to provide Clarion for Windows developers with state-of-the-art addons to increase your productivity, ease development and make it a pleasure for your end users to use your applications. We are always striving to produce the cleanest, problem free, software possible and work closely with users in an ongoing effort to squash bugs when they are found."

Posted on February 15, 1999

[The Handy Tools Page \(Gus Creces\)](#)

DiskFunctions, CopyImages, IniFile, VblFileNames, ApplicationImages, EmbedInformation, LimitExportClasses, DontExportHandy, LocatorOverrideControl, Internet FTP. Try Before You Buy.

Posted on February 15, 1999

[SoftStuff](#)

We will soon have a whole selection of utilities and plug in apps for developers. The majority of apps will be compatible with Clarion, but some will be totally stand alone. This an app that generates data for demo purposes however you may add or change as you wish.

Posted on March 3, 1999

[Princen Information Technology](#)

Templates to directly send internet mail, control user prompts, play multimedia files, print dictionary, show directory listbox, limit threads, print utilities, DDE to control Word/Excel.

Posted on March 27, 1999

[InFranView Freeware Graphic Viewer](#)

IrfanView is a very fast FREEWARE (for non-commercial use) 32-Bit graphic viewer for Windoze95/98 and Windows NT. Commercial registration is US\$10, source is available for an additional fee. Recommended by Clarion developer Nick LeForte.

Posted on April 2, 1999

[LogicBit Programmer's Power Tool Library](#)

LOGICBit creates hardware level multimedia tools (MMLib32) and has just announced the Clarion programmer's power tool library XLib32. XLib32 includes support for Text-to-Speech, TTY, TTY-to-Speech, 4 low level compression algorithms, 4 128 bit encryption algorithms and lot's more. Since LOGICBit tools are not templates you can rest assured that all LOGICBit tools work seamlessly in all version of windows.

Posted on April 9, 1999

Knowledge Bases

[The CW SuperPage](#)

"The CW SuperPage web site was an idea we had when searching the web for Clarion information. We could find no site with all the information we needed, so we decided to create our own. In our Delphi days we would always look for new Delphi files and info on the Delphi SuperPage. We have always thought that was a great name for a web site, so we began working on the CW SuperPage web site about three months ago. There were several technical problems that caused the site's deployment to be delayed, but now the CW SuperPage is online and ready to go. Our web site offers the developers eXchange, and is updated frequently with the latest CW news, tips & tricks as well as other interesting Clarion information."

Posted on February 15, 1999

[Steve Parker's Place](#)

"Largest and most complex publicly available CWIC site. Houses directory of Clarion Developers in Chicago, index to every article on Clarion programming, database of Clarion sites worldwide, sample personnel and job posting applications."

Posted on February 15, 1999

[IceTips Knowledge Base](#)

CWIC-based KB

Posted on February 15, 1999

[ForKeeps Message Archiver](#)

Not only is ForKeeps Mail & Message Archiver a unique CW based commercial application - it's also used by a large number of CW developers for keeping track of TopSpeed related Forum and newsgroup messages.

Posted on March 15, 1999

Reporting

[Gitano Software](#)

RenTrack property management software, various third party tools including calendars, calculators, notes management, print options/preview,
Posted on February 15, 1999

[CW Addons \(Lodestar Software\)](#)

"Lodestar Software is proud to provide Clarion for Windows developers with state-of-the-art addons to increase your productivity, ease development and make it a pleasure for your end users to use your applications. We are always striving to produce the cleanest, problem free, software possible and work closely with users in an ongoing effort to squash bugs when they are found."
Posted on February 15, 1999

[Creative PC Solutions](#)

"Creative Reporting Tools: Designed to help Clarion developers of all experience levels produce professional quality reports easily! Versions available for 2003, C4, and C5."
Posted on February 15, 1999

[Locus Software](#)

You can download free templates for Clarion 2.00x and Clarion C4. The templates for C4 works with the normal Clarion and Clarion ABC template chain. The templates contain: Inplace edit of browse-box with validation and lookup. File manager for updating files after changes to the dictionary. Rich Edit Control to view and edit *.rtf files and memo-fields with Mail Merge. Query by Example view and report of all files in your application Report filter with an improved report generator. Popup toolbar, popup date and popup directory. Popup help generation. Page tree. And more.....
Posted on February 19, 1999

[Mitten Software](#)

"Mitten Software provides time-saving tools and services for Clarion Developers. We support all versions of Clarion Software, both Windows and DOS. Contact us for a wide variety of programming tools, templates, books and educational material. Ask for help with your Clarion Software projects. We provide project management, consulting and contract programming."
Posted on February 23, 1999

[Berthume Software and Consulting](#)

Addons, Electronic Forms Solutions, and Custom Software Development. Products include Powerrun to launch processes in foreground or background, PTools for printing reports from disk, PCL Tools, which lets you use PCL commands in a report, and Transform Professional which creates forms from scanned images.
Posted on March 3, 1999

Software Development Tools

[Carl Barnes Computer Consulting](#)

Home of the CW Assistant and Clarion Source Search Program. Carl is a Clarion developer in the Chicago area, and a member of Team Topspeed.
Posted on February 15, 1999

[Home of Le Schmoo](#)

Clarion Compile Manager: Batch compiler for clarion APPs and PRJs.
Clarion Class Browser: Browse standard class headers and source, including reference variable lookups and inheritance trees.
Posted on February 15, 1999

[Tom Ruby](#)

Experienced software developer available for contract work. Available for Download: WanTimes - Synchronize Netware file server clocks to NIST; CW Dict Utility - My CW dictionary utility. Copies field, table and key names to clipboard. Build SQL command file to create tables and indices. Of course it's in CW. Now in C5 ABC (should work in C4) ;photos.zip -

Demo program which goes with article in Clarion Online.
Posted on February 15, 1999

[Software By Ragazzi](#)

"We provide quality add-ons which will give your Clarion programs a more professional look while making the job of programming easier and faster. Our line of templates and libraries allow you to concentrate on developing an application specific to your needs and leaving the mundane aspects of programming to our product line."
Posted on February 15, 1999

[Nice Touch Solutions Inc.](#)

"Dictionary Assistant (DA) and Application Assistant (AA) are query and reporting tools designed to answer these and many more questions that we, as Clarion developers ask. Through the use of application Utility Wizards these tools create normalized databases representing your application and dictionary (Files, Fields, Keys and Relationships, Procedures, Windows, Reports, Controls, Templates and Database File Usage)."
Posted on February 15, 1999

[Dalby Source Printer](#)

"Dalby Source Printer for Clarion sets completely new standards for source printers. Even if you don't print your sourcecodes as a hardcopy of your projects, you will have endless gains from Dalby Source Printer for Clarion to get the general view of your source-code. You can print the whole source code or simply embedpoints with userdefined syntax highlightning. Dalby Source Printer is now freeware and can be used without charge."
Posted on February 15, 1999

[The Handy Tools Page \(Gus Creces\)](#)

DiskFunctions, CopyImages, IniFile, VblFileNames, ApplicationImages, EmbedInformation, LimitExportClasses, DontExportHandy, LocatorOverrideControl, Internet FTP. Try Before You Buy.
Posted on February 15, 1999

[CapeSoft](#)

File Manager 2, Secwin, EzHelp, Multi-Prj, WinEvent, Object Writer
Posted on February 15, 1999

[Author Direct Shareware](#)

ABCFree templates, free WinAPI Toolkit, source code formatter, form view mode template, calendar, etc.
Posted on March 12, 1999

[Linder Software](#)

"Linder Software develops and markets high performance data compression libraries (LSZip, LSPack) and installation systems for programmers working in Windows."
Posted on February 17, 1999

[Locus Software](#)

You can download free templates for Clarion 2.00x and Clarion C4. The templates for C4 works with the normal Clarion and Clarion ABC template chain. The templates contain: Inplace edit of browse-box with validation and lookup. File manager for updating files after changes to the dictionary. Rich Edit Control to view and edit *.rtf files and memo-fields with Mail Merge. Query by Example view and report of all files in your application Report filter with an improved report generator. Popup toolbar, popup date and popup directory. Popup help generation. Page tree. And more.....
Posted on February 19, 1999

[App-Ref Cross Reference Tool](#)

App-Ref is a cross reference, analysing and standardization tool for Clarion Windows Programmers. It will import Clarion TXD files, and uses a template to extract information from your Clarion application files, which are then imported to App-Ref. during the import process App-Ref searches the CLW files for where files, fields and variables are used. Each App-Ref Project consists of a Dictionary and one or more App files.

You can store upto 99,999 Projects and there is no limit on the number of App files in a Project. App-Ref is a cross reference, analysing and standardization tool for Clarion Windows Programmers. It will import Clarion TXD files, and uses a template to extract information from your Clarion application files, which are then imported to App-Ref. during the import process App-Ref searches the CLW files for where files, fields and variables are used. Each App-Ref Project consists of a Dictionary and one or more App files. you can store upto 99,999 Projects and there
Posted on February 22, 1999

[Mitten Software](#)

"Mitten Software provides time-saving tools and services for Clarion Developers. We support all versions of Clarion Software, both Windows and DOS. Contact us for a wide variety of programming tools, templates, books and educational material. Ask for help with your Clarion Software projects. We provide project management, consulting and contract programming."

Posted on February 23, 1999

[Berthume Software and Consulting](#)

Addons, Electronic Forms Solutions, and Custom Software Development. Products include Powerrun to launch processes in foreground or background, PTools for printing reports from disk, PCL Tools, which lets you use PCL commands in a report, and Transform Professional which creates forms from scanned images.

Posted on March 3, 1999

[Princen Information Technology](#)

Templates to directly send internet mail, control user prompts, play multimedia files, print dictionary, show directory listbox, limit threads, print utilities, DDE to control Word/Excel.

Posted on March 27, 1999

[ParseRat Data Extractor](#)

This utility can extract data from text, binary, dbase, and page image files, and format the data to text, dbase, Wordperfect, MS Word and other formats.

Posted on March 28, 1999

Security

[NextAge Consulting](#)

NextAge Consulting provides a full range of consulting services to our clients. This includes software development, website design and hosting, IT and project management, and joint ventures with other software developers. Most of our team of employees and subcontractors have been providing consulting services for 10 to 20 years in a wide range of industries and technologies. "Feedbackware" templates include application images, multicategory lookup, security, and miscellaneous.

Posted on February 15, 1999

[Gitano Software](#)

RenTrack property management software, various third party tools including calendars, calculators, notes management, print options/preview,

Posted on February 15, 1999

[Clarion Paranoid Home Page](#)

"Here I'll try to show you some tips how to secure your applications created in Clarion. My tips should work in any current version of Clarion for Windows (CW2003, C4, C5) using standard Clarion templates. I don't use ABC templates yet, but if you know it, you should easy apply this tips in your applications."

Posted on February 15, 1999

[Tinman Development Corp.](#)

DAS Registration, DAS Tools, DAS Security, DAS Quickhelp, Tintools (free), DAS tagging (free)

Posted on February 15, 1999

[CapeSoft](#)

File Manager 2, Secwin, EzHelp, Multi-Prj, WinEvent, Object Writer
Posted on February 15, 1999

[BoxSoft Development Inc.](#)

"Mike Hanson's Super Templates for Windows expands Clarion's Application Generation power. You'll develop higher quality programs that your clients will pay a premium for. Mike's Models and Templates for DOS have been popular for years, and now are available for Windows. Get the features you need. Mix and match these templates using Clarion's template registry." Many, many features, some free templates.
Posted on February 23, 1999

[Mitten Software](#)

"Mitten Software provides time-saving tools and services for Clarion Developers. We support all versions of Clarion Software, both Windows and DOS. Contact us for a wide variety of programming tools, templates, books and educational material. Ask for help with your Clarion Software projects. We provide project management, consulting and contract programming."
Posted on February 23, 1999

SQL & ODBC

[The CW SuperPage](#)

"The CW SuperPage web site was an idea we had when searching the web for Clarion information. We could find no site with all the information we needed, so we decided to create our own. In our Delphi days we would always look for new Delphi files and info on the Delphi SuperPage. We have always thought that was a great name for a web site, so we began working on the CW SuperPage web site about three months ago. There were several technical problems that caused the site's deployment to be delayed, but now the CW SuperPage is online and ready to go. Our web site offers the developers eXchange, and is updated frequently with the latest CW news, tips & tricks as well as other interesting Clarion information."
Posted on February 15, 1999

[NT Tools & Clarion Tips/Tricks](#)

Clarion 5 Tips and Tricks. Preview the latest MS-SQL Tips and Tricks for C5. API Call to Get the OS version. Hot-Keyed fields and Entry Color fields FEB99.ZIP. Handout. Sample objects are used. API call to capture the Win NT / 98 / 95 OS User Name.
Posted on February 15, 1999

[CW Addons \(Lodestar Software\)](#)

"Lodestar Software is proud to provide Clarion for Windows developers with state-of-the-art addons to increase your productivity, ease development and make it a pleasure for your end users to use your applications. We are always striving to produce the cleanest, problem free, software possible and work closely with users in an ongoing effort to squash bugs when they are found."
Posted on February 15, 1999

[Cowboy Computing Solutions](#)

SQL templates for Clarion
Posted on February 15, 1999

[RCLC Software](#)

Direct ODBC class for Clarion. RCLC Software is group of the Russian programmers, working in various organizations of St. Petersburg. Main direction of operations this creation on the Clarion language of application systems working in the strategy client-server. "As the standard tools of the Clarion language do not contain full the list of tools for operation with SQL databases, our group developed additional tools allowing to work more effectively. I think that some of these

developments will be useful to all developers using the Clarion language."

Posted on February 15, 1999

[The Network Connection](#)

Windows document imaging, Visual Voice Pro toolkit, and Steve Olensky's SQL Page.

Posted on March 7, 1999

[Pervasive I*net Data Server](#)

Pervasive's I*net Data Server (IDS) provides true client/server database interaction over the Internet or intranet. As a component of the Pervasive.SQL v7 family of products, IDS extends Pervasive.SQL applications and utilities to let you deploy current applications, anywhere, using the Internet as your network. Plus, your application scales seamlessly from standalone to client/server to the Internet³/₄ without any code changes, and with complete data security.

Posted on April 2, 1999

[Troy Sorzano's SQL FAQs](#)

SQL articles, tips, links, and even Troy's world renowned Stupid Temp Table Theory.

Posted on April 12, 1999

Miscellaneous

[NextAge Consulting](#)

NextAge Consulting provides a full range of consulting services to our clients. This includes software development, website design and hosting, IT and project management, and joint ventures with other software developers. Most of our team of employees and subcontractors have been providing consulting services for 10 to 20 years in a wide range of industries and technologies. "Feedbackware" templates include application images, multcategory lookup, security, and miscellaneous.

Posted on February 15, 1999

[Gitano Software](#)

RenTrack property management software, various third party tools including calendars, calculators, notes management, print options/preview,

Posted on February 15, 1999

[Logic Central](#)

Mike Pasley's site - "Logic Central is my web site where I'll have application engineering items that I find interesting and hope you find interesting as well. Of course, there is a focus on Clarion from TopSpeed Corporation. I run across things that helps me better use the Clarion development environment and I would like to share some of these items. So, I hope that this site presents to you useful concepts, old reminders, or new ideas that challenge you or stimulates discussion with others." Source for "Simplates"

Posted on February 15, 1999

[Unisoft FTP Site](#)

FTP site with a variety of information including DevCon '98 stuff.

Posted on February 15, 1999

[Sterling Data](#)

BackFlash - Data Back Up Module IMPEX - Import/Export templates Clarion Downloads MARKETEEER is a Contacts Manager and Sales Prospecting System - available in Windows and DOS versions - single user and for networks. Never miss a follow up! Keep control of your business and contacts! TIME LORD is a PIM (Personal Information Manager) to organize all your appointments, reminders, to do lists, phone calls and contacts. Available for DOS only. A demo is available for download. COURT MANAGER is designed to handle bookings and keep track of member details for squash and tennis clubs. DOS version only at the moment. BOOK STACK serves as a "study assistant" - notes you

have made while reading a book are stored with a reminder date. To ensure proper memorizing/revision these will pop up at predefined periods. Also stores your books and quotations collection - searchable by title, author, category etc. Only available in DOS version at present.
Posted on February 15, 1999

[Encourager Software](#)

Home of the Clarion Profile Exchange
Posted on February 15, 1999

[Tom Ruby](#)

Experienced software developer available for contract work. Available for Download: WanTimes - Synchronize Netware file server clocks to NIST; CW Dict Utility - My CW dictionary utility. Copies field, table and key names to clipboard. Build SQL command file to create tables and indices. Of course it's in CW. Now in C5 ABC (should work in C4) ; photos.zip - Demo program which goes with article in Clarion Online.
Posted on February 15, 1999

[Deccan Systems Inc.](#)

"DeccaPro - Intuitive, versatile and powerful cost estimating software for all your business needs." Written in Clarion.
Posted on February 15, 1999

[Software By Ragazzi](#)

"We provide quality add-ons which will give your Clarion programs a more professional look while making the job of programming easier and faster. Our line of templates and libraries allow you to concentrate on developing an application specific to your needs and leaving the mundane aspects of programming to our product line."
Posted on February 15, 1999

[J&S Software Co.](#)

" We have been writing custom business software since 1991. Our programming language of choice is Clarion for Windows. It is the finest business database language around. Along with the custom software development, we also write add-on tools for other Clarion for Windows (aka CW) developers. We are very proud of our Mailmerge add-on. Download the demo and check it out!" Producers of In-Back.
Posted on February 15, 1999

[NT Tools & Clarion Tips/Tricks](#)

Clarion 5 Tips and Tricks. Preview the latest MS-SQL Tips and Tricks for C5. API Call to Get the OS version. Hot-Keyed fields and Entry Color fields FEB99.ZIP. Handout. Sample objects are used. API call to capture the Win NT / 98 / 95 OS User Name.
Posted on February 15, 1999

[Andrew Guidroz II \(Official Cajun Chef of the Clarion Community\)](#)

"I have a lot of different interests. I program for a living using primarily Clarion for Windows from Topspeed Corporation. And I also LOVE to cook. All of us here in Louisiana put a strong emphasis on cooking -- especially crawfish. And being Cajun makes it all that much more special."
Posted on February 15, 1999

[The Handy Tools Page \(Gus Creces\)](#)

DiskFunctions, CopyImages, IniFile, VblFileNames, ApplicationImages, EmbedInformation, LimitExportClasses, DontExportHandy, LocatorOverrideControl, Internet FTP. Try Before You Buy.
Posted on February 15, 1999

[CWICWeb](#)

Quick Web - The information place for Clarion for Windows and Microsoft networking and operating systems certification. "A Web Site Dedicated to the CWIC side of Clarion for Windows 2.003,4,5EE "
Posted on February 15, 1999

[Tinman Development Corp.](#)

DAS Registration, DAS Tools, DAS Security, DAS Quickhelp, Tintools (free), DAS tagging (free)

Posted on February 15, 1999

[Unofficial IC Bug List](#)

Posted on February 15, 1999

[CapeSoft](#)

File Manager 2, Secwin, EzHelp, Multi-Prj, WinEvent, Object Writer
Posted on February 15, 1999

[Author Direct Shareware](#)

ABCFree templates, free WinAPI Toolkit, source code formatter, form view mode template, calendar, etc.
Posted on March 12, 1999

[Clarion Info For Pastel Partner Accounting Systems](#)

There is a discussion list for Clarion Programmers who write Add-ons for Pastel Partner Accounting Systems (<http://cwpastel.listbot.com>). Pastel's home page is at <http://www.pastel.com>.
Posted on February 16, 1999

[Linder Software](#)

"Linder Software develops and markets high performance data compression libraries (LSZip, LSPack) and installation systems for programmers working in Windows."
Posted on February 17, 1999

[Locus Software](#)

You can download free templates for Clarion 2.00x and Clarion C4. The templates for C4 works with the normal Clarion and Clarion ABC template chain. The templates contain: Inplace edit of browse-box with validation and lookup. File manager for updating files after changes to the dictionary. Rich Edit Control to view and edit *.rtf files and memo-fields with Mail Merge. Query by Example view and report of all files in your application Report filter with an improved report generator. Popup toolbar, popup date and popup directory. Popup help generation. Page tree. And more.....
Posted on February 19, 1999

[BoxSoft Development Inc.](#)

"Mike Hanson's Super Templates for Windows expands Clarion's Application Generation power. You'll develop higher quality programs that your clients will pay a premium for. Mike's Models and Templates for DOS have been popular for years, and now are available for Windows. Get the features you need. Mix and match these templates using Clarion's template registry." Many, many features, some free templates.
Posted on February 23, 1999

[Mitten Software](#)

"Mitten Software provides time-saving tools and services for Clarion Developers. We support all versions of Clarion Software, both Windows and DOS. Contact us for a wide variety of programming tools, templates, books and educational material. Ask for help with your Clarion Software projects. We provide project management, consulting and contract programming."
Posted on February 23, 1999

[Split Bar Region Resizing Template](#)

This template implements a simple "split-bar" type control that is common to many programs (like Windows 95 Explorer). This "split-bar" will allow your users to resize the regions, list boxes, etc. as they see fit. For those controls (buttons, etc.) that are dependent on the position of a particular region or list box, the template allows these controls to be moved instead of resized. FREE
Posted on February 24, 1999

[SilverGhost](#)

Some free templates, including one to read volume serial numbers.
Posted on February 26, 1999

[SoftStuff](#)

We will soon have a whole selection of utilities and plug in apps for developers. The majority of apps will be compatible with Clarion, but some will be totally stand alone. This an app that generates data for demo purposes however you may add or change as you wish.

Posted on March 3, 1999

[Berthume Software and Consulting](#)

Addons, Electronic Forms Solutions, and Custom Software Development. Products include Powerrun to launch processes in foreground or background, PTools for printing reports from disk, PCL Tools, which lets you use PCL commands in a report, and Transform Professional which creates forms from scanned images.

Posted on March 3, 1999

[The Network Connection](#)

Windows document imaging, Visual Voice Pro toolkit, and Steve Olensky's SQL Page.

Posted on March 7, 1999

[Princen Information Technology](#)

Templates to directly send internet mail, control user prompts, play multimedia files, print dictionary, show directory listbox, limit threads, print utilities, DDE to control Word/Excel.

Posted on March 27, 1999

[XBS Object Server](#)

XBS - Server Multi Tasks your Business Objects and ABC Objects. This is like an an OOP Object Runtime OS for handling the execution of Object processes real time.

Posted on April 3, 1999

[LogicBit Programmer's Power Tool Library](#)

LOGICBit creates hardware level multimedia tools (MMLib32) and has just announced the Clarion programmer's power tool library XLib32. XLib32 includes support for Text-to-Speech, TTY, TTY-to-Speech, 4 low level compression algorithms, 4 128 bit encryption algorithms and lot's more. Since LOGICBit tools are not templates you can rest assured that all LOGICBit tools work seamlessly in all version of windows.

Posted on April 9, 1999

Clarion Information

[Logic Central](#)

Mike Pasley's site - "Logic Central is my web site where I'll have application engineering items that I find interesting and hope you find interesting as well. Of course, there is a focus on Clarion from Topspeed Corporation. I run across things that helps me better use the Clarion development environment and I would like to share some of these items. So, I hope that this site presents to you useful concepts, old reminders, or new ideas that challenge you or stimulates discussion with others." Source for "Simplates"

Posted on February 15, 1999

[Unisoft FTP Site](#)

FTP site with a variety of information including DevCon '98 stuff.

Posted on February 15, 1999

[Sterling Data](#)

BackFlash - Data Back Up Module IMPEX - Import/Export templates Clarion Downloads MARKETEEER is a Contacts Manager and Sales Prospecting System - available in Windows and DOS versions - single user and for networks. Never miss a follow up! Keep control of your business and contacts! TIME LORD is a PIM (Personal Information Manager) to organize all your appointments, reminders, to do lists, phone calls and contacts. Available for DOS only. A demo is available for download. COURT MANAGER is designed to handle bookings and keep track of member details for squash and tennis clubs. DOS version only at the moment. BOOK STACK serves as a "study assistant" - notes you

have made while reading a book are stored with a reminder date. To ensure proper memorizing/revision these will pop up at predefined periods. Also stores your books and quotations collection - searchable by title, author, category etc. Only available in DOS version at present.
Posted on February 15, 1999

[ProDomus Inc.](#)

CWIntl -- International Localization, PD Translator International Localization, PAddress -- Worldwide Address Formats, PD Lookup -- Better Lookups: PDLookup, browse button lookup. PD Drops drop lookups. PD Date Tools - Scrolling date objects, date functions, time zone information, and threaded calendar. Localized date function rules, month names and day names. PDFin - Finance Toolkit Simple Tools -- Article and source code example about developing 16 and 32-bit libraries. Includes modified mangle program and source code (prj, exp, clw, inc, and tpl files)
Posted on February 15, 1999

[Australia's Clarion WWW Site](#)

CUGQ Clarion User Group Queensland
SCUG Sydney Clarion User Group
SACUG South Australia Clarion User Group
ClaSIG(VCUG) Victorian Clarion User Group
Other International Clarion User Groups
A VERY Large collection of Clarion related links on the WWW, over 150 at last count. Clarion.org.au FTP Site - Files from Sable Software BBS collection, Lots of Clarion 2.x and 3x DOS files (Under construction at present)
Posted on February 15, 1999

[Encourager Software](#)

Home of the Clarion Profile Exchange
Posted on February 15, 1999

[Clarion Paranoid Home Page](#)

"Here I'll try to show you some tips how to secure your applications created in Clarion. My tips should work in any current version of Clarion for Windows (CW2003, C4, C5) using standard Clarion templates. I don't use ABC templates yet, but if you know it, you should easy apply this tips in your applications."
Posted on February 15, 1999

[The CW SuperPage](#)

"The CW SuperPage web site was an idea we had when searching the web for Clarion information. We could find no site with all the information we needed, so we decided to create our own. In our Delphi days we would always look for new Delphi files and info on the Delphi SuperPage. We have always thought that was a great name for a web site, so we began working on the CW SuperPage web site about three months ago. There were several technical problems that caused the site's deployment to be delayed, but now the CW SuperPage is online and ready to go. Our web site offers the developers eXchange, and is updated frequently with the latest CW news, tips & tricks as well as other interesting Clarion information."
Posted on February 15, 1999

[NT Tools & Clarion Tips/Tricks](#)

Clarion 5 Tips and Tricks. Preview the latest MS-SQL Tips and Tricks for C5. API Call to Get the OS version. Hot-Keyed fields and Entry Color fields FEB99.ZIP. Handout. Sample objects are used. API call to capture the Win NT / 98 / 95 OS User Name.
Posted on February 15, 1999

[Ross Santos](#)

Co-author of "Developing Clarion For Windows Applications"
Posted on February 15, 1999

[RCLC Software](#)

Direct ODBC class for Clarion. RCLC Software is group of the Russian programmers , working in various organizations of St. Petersburg. Main direction of operations this creation on the Clarion language of

application systems working in the strategy client-server. "As the standard tools of the Clarion language do not contain full the list of tools for operation with SQL databases, our group developed additional tools allowing to work more effectively. I think that some of these developments will be useful to all developers using the Clarion language."

Posted on February 15, 1999

[Steve Parker's Place](#)

"Largest and most complex publicly available CWIC site. Houses directory of Clarion Developers in Chicago, index to every article on Clarion programming, database of Clarion sites worldwide, sample personnel and job posting applications."

Posted on February 15, 1999

[Unofficial IC Bug List](#)

Posted on February 15, 1999

[Clarion Info For Pastel Partner Accounting Systems](#)

There is a discussion list for Clarion Programmers who write Add-ons for Pastel Partner Accounting Systems (<http://cwpastel.listbot.com>). Pastel's home page is at <http://www.pastel.com>.

Posted on February 16, 1999

[The DAB Files](#)

David Bayliss is the Software Development Manager for Topspeed Corp., where his job is "running a small and enthusiastic department of software developers, producing the worlds best software development tools, period." Clarion-related articles, links.

Posted on February 19, 1999

[Mitten Software](#)

"Mitten Software provides time-saving tools and services for Clarion Developers. We support all versions of Clarion Software, both Windows and DOS. Contact us for a wide variety of programming tools, templates, books and educational material. Ask for help with your Clarion Software projects. We provide project management, consulting and contract programming."

Posted on February 23, 1999

Consultants

[Carl Barnes Computer Consulting](#)

Home of the CW Assistant and Clarion Source Search Program. Carl is a Clarion developer in the Chicago area, and a member of Team Topspeed.

Posted on February 15, 1999

[Top3systems Corporation](#)

This site is still under construction.

Posted on February 15, 1999

[NextAge Consulting](#)

NextAge Consulting provides a full range of consulting services to our clients. This includes software development, website design and hosting, IT and project management, and joint ventures with other software developers. Most of our team of employees and subcontractors have been providing consulting services for 10 to 20 years in a wide range of industries and technologies. "Feedbackware" templates include application images, multicategory lookup, security, and miscellaneous.

Posted on February 15, 1999

[Gitano Software](#)

RenTrack property management software, various third party tools including calendars, calculators, notes management, print options/preview,

Posted on February 15, 1999

[Sterling Data](#)

BackFlash - Data Back Up Module IMPEX - Import/Export templates
Clarion Downloads MARKETEEER is a Contacts Manager and Sales
Prospecting System - available in Windows and DOS versions - single
user and for networks. Never miss a follow up! Keep control of your
business and contacts! TIME LORD is a PIM (Personal Information
Manager) to organize all your appointments, reminders, to do lists,
phone calls and contacts. Available for DOS only. A demo is available for
download. COURT MANAGER is designed to handle bookings and keep
track of member details for squash and tennis clubs. DOS version only at
the moment. BOOK STACK serves as a "study assistant" - notes you
have made while reading a book are stored with a reminder date. To
ensure proper memorizing/revision these will pop up at predefined
periods. Also stores your books and quotations collection - searchable by
title, author, category etc. Only available in DOS version at present.
Posted on February 15, 1999

[Tom Ruby](#)

Experienced software developer available for contract work. Available for
Download: WanTimes - Synchronize Netware file server clocks to NIST;
CW Dict Utility - My CW dictionary utility. Copies field, table and key
names to clipboard. Build SQL command file to create tables and indices.
Of course it's in CW. Now in C5 ABC (should work in C4) ;photos.zip -
Demo program which goes with article in Clarion Online.
Posted on February 15, 1999

[Software By Ragazzi](#)

"We provide quality add-ons which will give your Clarion programs a
more professional look while making the job of programming easier and
faster. Our line of templates and libraries allow you to concentrate on
developing an application specific to your needs and leaving the
mundane aspects of programming to our product line."
Posted on February 15, 1999

[CW Addons \(Lodestar Software\)](#)

"Lodestar Software is proud to provide Clarion for Windows developers
with state-of-the-art addons to increase your productivity, ease
development and make it a pleasure for your end users to use your
applications. We are always striving to produce the cleanest, problem
free, software possible and work closely with users in an ongoing effort
to squash bugs when they are found."
Posted on February 15, 1999

[Mitten Software](#)

"Mitten Software provides time-saving tools and services for Clarion
Developers. We support all versions of Clarion Software, both Windows
and DOS. Contact us for a wide variety of programming tools, templates,
books and educational material. Ask for help with your Clarion Software
projects. We provide project management, consulting and contract
programming."
Posted on February 23, 1999

Web Design

[Gitano Software](#)

RenTrack property management software, various third party tools
including calendars, calculators, notes management, print
options/preview,
Posted on February 15, 1999

[Sterling Data](#)

BackFlash - Data Back Up Module IMPEX - Import/Export templates
Clarion Downloads MARKETEEER is a Contacts Manager and Sales
Prospecting System - available in Windows and DOS versions - single
user and for networks. Never miss a follow up! Keep control of your
business and contacts! TIME LORD is a PIM (Personal Information
Manager) to organize all your appointments, reminders, to do lists,
phone calls and contacts. Available for DOS only. A demo is available for

download. COURT MANAGER is designed to handle bookings and keep track of member details for squash and tennis clubs. DOS version only at the moment. BOOK STACK serves as a "study assistant" - notes you have made while reading a book are stored with a reminder date. To ensure proper memorizing/revision these will pop up at predefined periods. Also stores your books and quotations collection - searchable by title, author, category etc. Only available in DOS version at present. Posted on February 15, 1999

[Software By Ragazzi](#)

"We provide quality add-ons which will give your Clarion programs a more professional look while making the job of programming easier and faster. Our line of templates and libraries allow you to concentrate on developing an application specific to your needs and leaving the mundane aspects of programming to our product line." Posted on February 15, 1999

[CW Addons \(Lodestar Software\)](#)

"Lodestar Software is proud to provide Clarion for Windows developers with state-of-the-art addons to increase your productivity, ease development and make it a pleasure for your end users to use your applications. We are always striving to produce the cleanest, problem free, software possible and work closely with users in an ongoing effort to squash bugs when they are found." Posted on February 15, 1999

Internet-Related

[The Handy Tools Page \(Gus Creces\)](#)

DiskFunctions, CopyImages, IniFile, VblFileNames, ApplicationImages, EmbedInformation, LimitExportClasses, DontExportHandy, LocatorOverrideControl, Internet FTP. Try Before You Buy. Posted on February 15, 1999

[CWICWeb](#)

Quick Web - The information place for Clarion for Windows and Microsoft networking and operating systems certification. "A Web Site Dedicated to the CWIC side of Clarion for Windows 2.003,4,5EE " Posted on February 15, 1999

[The All-Clarion Web Server](#)

"It's entirely written in Clarion It uses direct Winsock API calls. It's running on Windows 98 (NT later), it is asynchronous, and adheres to HTTP 1.0 protocol. It demonstrates what you can do, totally in Clarion. Please look for updated announcements at Logic Central (<http://www.logiccentral.com>). Please note, that Igor Kuklin has fine example of a complete Web server implementation submitted at the Icetips (<http://www.icetips.com>) website." Posted on February 18, 1999

[Princen Information Technology](#)

Templates to directly send internet mail, control user prompts, play multimedia files, print dictionary, show directory listbox, limit threads, print utilities, DDE to control Word/Excel. Posted on March 27, 1999

[Pervasive I*net Data Server](#)

Pervasive's I*net Data Server (IDS) provides true client/server database interaction over the Internet or intranet. As a component of the Pervasive.SQL v7 family of products, IDS extends Pervasive.SQL applications and utilities to let you deploy current applications, anywhere, using the Internet as your network. Plus, your application scales seamlessly from standalone to client/server to the Internet without any code changes, and with complete data security. Posted on April 2, 1999

[Catalyst SocketTools Examples](#)

This site contains the following examples of using Catalyst's SocketTools

Library Edition: FTP - performs asynchronous FTP download; SMTP - email delivery with optional attachment; Sockets - simple Echo session using TCP sockets.

Posted on May 17, 1999

Linux And Clarion

[Open Linux Project For TopSpeed](#)

Goals: We create templates that can take existing Clarion .app applications and translate them into code that can be compiled into Linux programs. We create templates that allow us to use our familiar Clarion IDE to generate Linux programs. We create file drivers that allow us to access common data files, so that either Windows or Linux can access the same data. We work together or separately but keep all source in a common location to be accessed by everyone. We keep focused but not be rigid in our views. We accept all feedback and/or suggestions. We select people that will keep us going in the right direction. We solicit TopSpeed in the project. We do not require them to give up valuable resources, but only ask for support/suggestions when we get into a bind. We hope that TopSpeed with support us in this endeavour.

Posted on February 22, 1999

Distributors

[Advantage Software BV](#)

Advantage Software BV is the distributor and supportcenter of TopSpeed products in the Benelux (The Netherlands, Belgium and Luxembourg). The activities at Advantage are sales in TopSpeed products such as Clarion for Windows/Dos, TopSpeed C++ and lots of Accessories for Clarion products. Also custom-made projects (software development), support, education and consultancy.

Posted on March 12, 1999

[TopSpeed International Distributors](#)

Complete list of international distributors of TopSpeed products.

Posted on March 12, 1999

Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Feature Article

The ABCs of OOP

Understanding Inheritance and Encapsulation

by David Harms

In the [first article in this series](#) I reviewed the basic concepts of object-oriented programming, including inheritance, encapsulation, polymorphism, and composition. In this article I'll focus on inheritance and encapsulation, which will lay a foundation for the critical area of virtual methods.

Inheritance is a terrific idea, and not just if you have a rich uncle. The more code you've written in your life, the more times you've probably written the same code over and over, only slightly differently each time. Clarion has traditionally used its template technology to take care of this problem, generating oft-used code for you and tailoring it to your needs. This doesn't exactly reduce the code being created, but it does make the programmer's job a bit easier.

Inheritance uses a different approach than the templates to code reuse (although in Clarion the ABC templates and classes both contribute vital pieces of the solution). Rather than rewriting the code with your changes, inheritance allows you to selectively replace or supplement the original code.

A Derived Class

Previously I demonstrated a [small class](#) which could be used to assist in debugging applications. Listing 1 shows the class declaration and Listing 2 the class implementation.

Listing 1. The DebugClass declaration (DEBUG.INC).

```
TraceQueue    QUEUE,TYPE
Text          STRING(200)
              END

DebugClass    CLASS,TYPE,MODULE('DEBUG.CLW')
TraceQ        &TraceQueue
Construct     PROCEDURE
Destruct      PROCEDURE
ShowTrace     PROCEDURE
Trace         PROCEDURE(STRING Text)
              END
```

etc
2000

If you're interested
take our poll &
let us know!

BKO
Enterprises, Inc.

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

Listing 2. The DebugClass implementation (DEBUG.CLW).

```

MEMBER

    MAP
    END

    INCLUDE( 'DEBUG.INC' )

DebugClass.Construct          PROCEDURE
    CODE
    SELF.TraceQ &= NEW(TraceQueue)

DebugClass.Destruct          PROCEDURE
    CODE
    FREE(SELF.TraceQ)
    DISPOSE(SELF.TraceQ)

DebugClass.ShowTrace          PROCEDURE

window WINDOW('Debug Messages'), AT(, , 331, 206), |
    FONT('MS Sans Serif', 8, , , CHARSET:ANSI), SYSTEM, GRAY, DOUBLE
    LIST, AT(5, 5, 320, 180), USE(?List1), HVSCROLL, |
    FONT('Courier New', 8, , , FONT:regular, CHARSET:ANSI), |
    FROM(self.TraceQ)
    BUTTON('Close'), AT(150, 190, , 14), USE(?Close)
    END

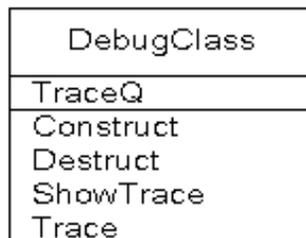
    CODE
    OPEN(WINDOW)
    ACCEPT
        IF FIELD() = ?Close AND EVENT() = EVENT:Accepted
            BREAK
        END
    END

DebugClass.Trace              PROCEDURE(STRING Text)
    CODE
    SELF.TraceQ.Text = Text
    ADD(SELF.TraceQ)

```

This class can also be diagrammed as shown in Figure 1.

Figure 1. The DebugClass class diagram.



The class notation used in Figure 1 is fairly standard. The class is drawn as a box with the top area containing the name. Below it are two optional areas. The first holds properties (in this case the queue called `TraceQ`) and below that are the methods. I'll be using this kind of class diagram in explaining inheritance.

Let's say you've been using the `DebugClass` to create a trace log of messages while debugging one of your applications. You now decide that you want to modify the format of the trace log to display a timestamp at the beginning of each line. One possibility is to add `FORMAT(CLOCK(),@T4) & ' '` to the beginning of the line of text. But if you make that change to `DebugClass` then you've altered your original code, and there still may be times when you don't want the timestamp.

If you sometimes want a feature in your code and other times don't, procedural code gives you two options. One is to add a switch to the code so you can turn the feature on and off. The other is to make a complete copy of the code, then change the copy. The latter is a fairly ugly solution to this problem but when you're dealing with more complex logic it's sometimes the only solution.

In OOP you can solve this problem with inheritance. Simply create a class derived from `DebugClass` and replace the `Trace` method with the one you want to use. Listing 3 shows a declaration for a derived `DebugTimedClass`, and Listing 4 shows the method source (or one possible version thereof).

Listing 3. The `DebugTimedClass` declaration (`DEBUGTIMED.INC`).

```
include('debug.inc')

DebugTimedClass CLASS(DebugClass),TYPE,MODULE('DEBUGTIMED.CLW')
Trace          PROCEDURE(String Text)
END
```

`DebugTimedClass` is now the child of `DebugClass`, and `DebugClass` is the parent of `DebugTimedClass`. Children are derived from their parents (you knew that).

Note that the form of the class declaration has now changed. Whereas `DebugClass` had a `CLASS` attribute with no parameters, meaning it was a base class, `DebugTimedClass` has the `CLASS(DebugClass)` attribute. This means that `DebugTimedClass` is identical to `DebugClass` except where `DebugTimedClass` contains replacement or new methods. In this case you can see that there is a `Trace` method which has the identical declaration to that in `DebugClass`. The result is that `DebugTimedClass` will be have exactly as if you had created it by making a copy of the `DebugClass` declaration and implementation (`.INC` and `.CLW` files) and replacing the `Trace` method.

As with `DebugClass`, the declaration is kept in an include file, and that file also needs an `INCLUDE` statement that points back to `DEBUG.INC` so that the compiler knows where to find the parent class definition.

A method in a derived class which has the same name and parameter list as a method in the parent class is said to override the parent method. Overriding is just an OOPish term for replacing.

NOTE: Although you can override methods in Clarion you cannot override data. Variables and structures declared in a derived class may not have the same name as those in the parent class.

`DebugTimedClass`'s `MODULE` statement is also different from that of `DebugClass`: it points to the `DebugTimed.CLW` module which contains the source for the new `Trace` method. (You can have multiple classes in the same module if you wish, though this raises some scoping issues. More

on that later.)

Inheriting code is considerably more efficient than copying code because it avoids source duplication, leaving you only one place where you need to modify/maintain any given block of source code. It can, however, make the code slightly more difficult to read, in that you need to know what code is in the parent class. A [class browser](#) can be a big help when you're working with derived classes.

Listing 4 shows a possible implementation of the new `Trace` method.

Listing 4. A `Trace` method with a timestamp.

```
DebugTimedClass.Trace      procedure(string Text)
  code
  self.TraceQ.Text = clock() & ' ' & Text
  add(self.TraceQ)
```

The `Trace` method in Listing 4 will work, but it isn't as efficient as it could be. It's an almost complete duplication of the original `Trace` method except that it adds the timestamp to the log string, so if the name or structure of `TraceQ` should change, you'd have to change the code in the derived class as well. Listing 4 also assumes that `TraceQ`, which is a property of the parent class, can be accessed by the derived class, and there may be good reasons why this shouldn't be allowed to happen.

It so happens that in this case `TraceQ` is public data and can be accessed by the derived class, so there won't be a problem. But often a class will restrict access to some of its data so that it can guarantee the state of that data.

Restricting Data Access

One of the goals you should be aiming for in OOP is to make each class as self-contained (or encapsulated) as possible. That means that not only does the class have everything it needs to function (it doesn't rely on global data, for instance), it also means that outside forces can't make the class do something it shouldn't do. And usually the way you force a class to do something it shouldn't do is by changing some of its data (from outside the class) to values that the class can't properly deal with.

Some class data can safely be exposed. If you have a `BYTE` field which is used as a `True/False` flag, and your code consistently interprets a value of zero as `False` and all other values as `True`, then it doesn't matter what a programmer (or user) sets the flag to because your class can deal with the situation. If on the other hand you have a string field which can only contain the values `YES`, `NO`, `SOMETIMES` or `MAYBE` then you don't want anyone or anything setting that field to a value like `HARDLY EVER`. You do want to mediate access to the variable through one or more methods.

TIP: It's a convention in Clarion OOP to prefix a method name that gets a variable's content with `Get` and a method that sets a variable's content with `Set`.

The same principles apply to methods. You may have some methods that are only for the class's own use, and others that can be safely used by derived classes or other code.

You can restrict access to data and methods by using the `PRIVATE` and `PROTECTED` attributes.

PRIVATE And PROTECTED

The `PRIVATE` attribute means that the method or property can only be used by the class in which the method or property is declared. Use this attribute when you don't want any derived class or any other code whatsoever to be able to use a method or property. (There is one

exception to this rule – classes declared in the same source file can use all of each others methods and properties. This is Clarion's version of "friends," or what David Bayliss calls "controlled encapsulation leakage.")

If for instance the `TraceQ` property in `DebugClass` were private, the code in Listing 4 would result in a compiler error because the derived class wouldn't have access to the queue.

If you want a property or method to be available to only derived classes, use the `PROTECTED` attribute.

TIP: One way to assess your class design is to put the `PRIVATE` attribute on all your class data, changing to `PROTECTED` where necessary to allow derived classes to compile. This isn't a hard and fast rule, but generally the more `PRIVATE/PROTECTED` data you have in a class, the better the design because any access to the data is mediated by methods which can control the actual values that may be set.

Asking Your Parents

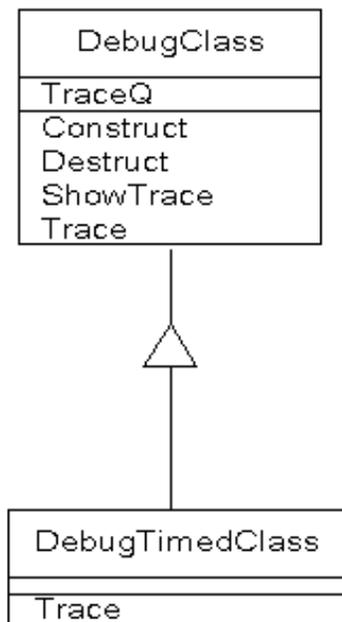
Even though `DebugClass`'s `TraceQ` is public, the code in Listing 4 isn't a great idea because it duplicates existing code, and one of the aims of OOP is to improve code reuse, not provide more opportunities for duplication. In Listing 5 the reference to `TraceQ` is replaced by a call to the parent class's trace method.

Listing 5. A better Trace method.

```
DebugTimedClass.Trace          PROCEDURE (STRING Text)
CODE
PARENT.Trace(CLOCK() & ' - ' & Text)
```

Just as `SELF` refers to the current object/class (see the [previous article](#) in this series), the `PARENT` keyword refers to the parent class. Figure 2 diagrams the relationship between the two classes.

Figure 2. A diagram showing the parent and child classes.



The triangle between the two classes shows the parent/child relationship. The top of the triangle is linked to the parent, and the base is linked to one or more child classes. Although you appear to be dealing with just one class in your code (`DebugTimedClass`) you're actually working with a composite of the base and derived classes. This is one of the things that can make OOP code look confusing at first. The class you're dealing with, if a derived class, contains much more code than is immediately apparent. Every method that's available in the parent class can be called from the derived (child) class, and every non-private property in the parent class can be used by the child class.

This doesn't mean that you need to use `PARENT` whenever you're working with a parent class method or variable. If a method exists in only the parent class then the compiler knows you mean `PARENT` even if you don't specify it. And if the method exists in both the parent and derived class, then the compiler always assumes you mean the derived method unless you use the `PARENT` keyword.

Listing 5 is a considerably more efficient design than is Listing 4 because it introduces the minimum amount of new code. As well, you can now make `TraceQ` private, and `DebugTimedClass` will still compile because it doesn't access the queue directly.

The end result of all of this is a small class which extends the functionality of the `Trace` method. That may seem like a lot of work for little effect, but then this is just a simple demonstration. In ABC, for example, there is a base class that defines a number of methods that are used to control the core behaviour of edit-in-place (EIP) controls. As new EIP controls are introduced (spin boxes, drop lists etc) classes are derived from the base EIP class, and these new classes add the unique functionality the control requires while making use of all of the common code.

Summary

Inheritance and encapsulation together make it possible to design self-contained blocks of code (classes) which efficiently reuse code. As a result the amount of generated code in ABC applications is a small fraction of that generated for legacy applications. ABC procedures implement classes which are derived from ABC classes, and which add just the code needed to perform procedure-specific tasks.

The kind of inheritance I've described in this article is what I think of as simple inheritance. One class is derived from another and adds some functionality. But there's much more to inheritance than this. Another much more powerful aspect is something called virtual methods, without which the combination of the ABC classes and templates simply would not work. That's coming up in the next installment.

Note: The debugging class used in this article is a rudimentary form of the [cciDebugClass](#) which is part of the Clarion Open Source Project and available by [download](#).

An example application with a working `DebugClass` is also available for [download](#).

[David Harms](#) is an independent software developer and the co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995). His company, CoveComm Inc, publishes Clarion Magazine.

Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Feature Article

Inside ABC – The FileManager

by David Bayliss

To understand the design strategy behind the `FileManager` you first need to understand the name. So far the classes I've covered in this series ([ErrorClass](#), [ConstantClass](#) and [FieldPairsClass](#)) all have names which end with "Class." That is to denote that they are logically complete entities in their own right. Clearly they are implemented using features of the Clarion language, but it is the class itself that logically provides the functionality.

When I came around to looking at the file part of the ABC structure I quickly realised that here was a very different problem. The largest bulk of file functionality already existed in the language in the form of the `FILE` structure support by the file drivers. In fact the file drivers can be viewed as objects (especially as Scott was keen to extend the file property syntax to expose some of the file driver data structures).

The file drivers have two major drawbacks when viewed from an ABC perspective:

1. They deal with files. At the template level Clarion files are uniquely gifted with all sorts of useful information supplied by the dictionary (including default values, validation, prompts, descriptions etc). The file drivers only (or mostly) support the information that is actually provided "down at the metal" of the file structure itself. Thus, if the whole of ABC were to use the file drivers directly, all of the value of having a data rich dictionary would be lost to the class hierarchy.
2. They are a black box. A potential beauty of the Clarion language is the way that the grunge of file access is hidden, thereby providing a nice clean programmer interface. The problem is that the wrapper prevents the underlying functionality from being readily extended or overridden.

From these problems arose the idea of a `FileManager` class. Essentially each file would be owned by a `FileManager`. The manager would provide a wrapper around the file entity, and the wrapper would embellish the functionality of the file by utilising (and acting as a repository for) the data about the file contained within the dictionary.

Future Proofing

The next aspect to consider is that the black box approach to file classes clearly cannot last for ever. If the ABC system really is to become a fully integrated development solution then, over time, it has to extend throughout the run-time system. The advantages to be gained by having a consistent object-oriented library that allows modification and expansion at all stages in the hierarchy are too compelling to be ignored. In fact, as Bruce Barrington announced at DevCon '98, the `FileClass` (note the name switch) project is under way and will be providing a major thrust (and raft of benefits) to our future products.

NextAge
Consulting
Imaging
Templates

Add Scanning and Document Storage to your app in 10 minutes

Only \$149

BKO
Enterprises, Inc.

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

etc
2000

If you're interested
take our poll &
let us know!

But this presents another problem. We have to be able to produce a `FileManager` interface now that will still be valid and supportable when the file drivers have changed completely. This is more challenging than the normal OOP problem. All classes need to encapsulate themselves, but the `FileManager` has to provide encapsulation for the file driver that is not in itself encapsulated. This is rather like the distinction between erecting a fence that will remind you not to walk on a flower bed and erecting a fence that will persuade an exuberant Labrador retriever not to walk on a flower bed. In particular it means that the `FileManager` has a higher proportion of `PRIVATE` and non-virtual methods than most of the other classes.

Static Usage

Another peculiarity of the `FileManager` (shared with the `RelationManager`) is that the instances of the class to be created will (at least by default) be static. Remember that the `FileManager` is really just the extension we would like to have made to the `FILE` itself (but couldn't). Thus it would have the same scope and persistence as a file buffer, i.e. global. But it's not quite global. Most files are actually threaded variables. This gives the `FileManager` a bit of a headache since some of the data it stores is global (the file name, for example, is the same for all threads), but some of it is threaded (the open state of a file is thread specific). Therefore it was decided to build thread management into the `FileManager` so that it could control inter-thread resources directly.

Aims & Issues

In tackling a class of the size and complexity of this one, it is worth splitting out and enumerating some of the tasks and requirements it is to meet.

1. Structure Storage. To provide services to return the keys of a file, fields of a file, number of components of a key etc.
2. File Snapshotting. A problem of having global buffers is that when some other procedure wants to use a file buffer it tends to corrupt the buffer contents. In the classic template chain, for example, the RI code actually corrupts buffer contents. A requirement of the manager class was to allow file buffers (and file state) to be snap-shotted and restored at a later date.
3. Auto-Increment. The manager is to provide support for automatically incrementing key components.
4. Retrieval and Update of records. This is fairly obvious.
5. Initialisation and validation. Whilst this is again fairly easy to specify it is one of the key capabilities of the Clarion data dictionary. `FileManager` must exploit these capabilities.

As well as providing some raw functionality there were some other issues and subtleties that significantly extended the work of implementation.

Firstly, the `FileManager` was to be smart and reliable. If you told it to open a file then it should go open the file, and do everything within its power to get the file open, including retrying, creating, building keys etc. The reason is to simplify the usage from within hand-code (or embed code). Because the methods will do all they possibly can to make the thing happen you can code as if it has happened. This meant the `FileManager` needed a tightly integrated `ErrorClass`. There was a problem, however, because sometimes you don't actually want the system to scream blue murder if it doesn't succeed. The fact the class failed is all the information you require. So most of the methods have "Try" equivalents, i.e. the `Open` method has a corresponding `TryOpen` method. The Try means that the method simply returns an error code rather than taking action on its own (such as presenting an error message).

Another subtlety is aliases. In Clarion aliases are treated pretty much as separate files. Whilst this is good from the IDE end of things it makes some of the object orientation horrible if files are mapped one-to-one with `FileManagers`.

Suppose the `FileManager` provided a virtual method to initialise a record into which you drop some embed code. Then whenever you create a record your code gets executed, unless of course you happen to be using an alias, which results in an integrity problem. Now you need to embed the code for every alias as well. Bleh! So we decided to get creative. The `FileManager` for an alias class contains a reference to the `FileManager` for the "real" file. Then whenever the alias `FileManager` is called it simply recalls on to the actual `FileManager` to do the work. Of course to do this we need to make frequent use of the File snapshotting technology.

The final tweak, one which came up fairly late and which is a mixed blessing, is the notion of `LazyOpen`. The idea here is that if you use a lot of files in a procedure it is better to open them one by one when required than open them all in a burst at the head of the procedure (which can cause a delay in SQL). This is no problem for people using ABC (as the ABC classes always know if they are about to touch a file) but it causes a bit of a headache for people doing hand-code as they need to warn the ABC system (using the `UseFile` method) that they are about to drill underneath the ABC layer. (Incidentally, this is one of the problems that will go away once the `FileClass` is implemented).

Initialisation

Originally the initialisation of the `FileManager` was very complex. The template generated code called many `Addxxxx` routines building up within the `FileManager` a replica of the data stored within the file driver itself. From beta three onwards the file drivers were extended so that the information within them was more readily accessible. For this reason the initialisation interface withered down to two genuine methods, both used to tie the `FileManager` into the outside world.

I have added two others, `Kill` and `SetThread`, as they are both tied to `FileManagerManagement` rather than `FileManager` itself.

```
FileManager.Init PROCEDURE(File File,ErrorClass E)
```

There is a one-one correspondence between instances of the `FileManager` and files used in the program. The `File` parameter sets up this mapping. The `ErrorClass` defaults to the global error manager (when the template generated init code is called). The advantage of having this reference as protected is that a given file (or batch of files) can have a different error manager installed to allow (say) less extreme responses to a given set of files being missing.

The coding is fairly straight forward, the usual array of queues being `NEWed` and class variables being set.

The only real magic is in the `AddFileMapping` method. This is there to help with lazy open, or more specifically, to help when a view is used. The view driver can return a list of files that are used in a view, but this is not quite what we want. We want a list of `FileManagers`. The mapping code is thus there to provide a mapping from file references to `FileManager` references. The mapping takes advantage of the fact that both file references and `FileManager` references can be used as `LONGs`. The `Filemappings` will be able to die once the `File` class comes along, and are thus private. In fact they use Clarion's rather nice "in module" private technology to allow the private methods to be completely removed from the `.inc` file.

```
FileManager.AddKey PROCEDURE(KEY k,STRING Desc,BYTE AutoInc)
```

This is the one remaining `Add` method. It is required because there are two pieces of information stored within the dictionary and required by the `FileManager` that are not available to the `FILE` driver. The first is the textual description of each key, and the second is the auto-increment nature of the key. The `AutoInc` value specified is the component of the key which is auto-increment. Within the templates this will always be defined as the minor-most component of the key although this restriction is not imposed by the base classes. (That said, non-minor-most component auto-incrementation is much less tested than the more normal form!)

Whilst the method itself is of little strategic importance it does use a number of relatively new or advanced coding techniques so it is worth a little further investigation.

The principle job of `AddKey` is to form a new record in the `FileKeyQueue`. This is a private property. Because Clarion allows unresolved forward references (i.e. allows opaque types) the queue structure is not defined within the `.inc` file, only within `ABFILE.CLW`.

Assigning the key reference and description is fairly straightforward. Use of the description field as fixed length (rather than assigning a reference) is not as painful as you may think because Clarion queue buffers are compressed when stored in memory so any trailing spaces simply squash up.

The interesting code starts with this reference assignment:

```
self.keys.fields &= NEW KeyFieldQueue
```

This code is setting one of the fields of the `FileKeyQueue` to be a new instance of the `KeyFieldQueue`. That means there will be a `KeyFieldQueue` for every record of the `FileKeyQueue`. In other words this is a queue of queues. This is a very powerful technique, but it is also something to be careful of. In the `Kill` method for the `FileManager` we have to step through the `FileKeyQueue` freeing up each one of these sub-queues.

Next note that `SELF.HasAutoInc` is set if `AutoInc` is true. `HasAutoInc` is set on a per file basis, not per key. The information is really redundant since `HasAutoInc` could always be computed simply by stepping through the values in the key queue, but `HasAutoInc` is hit sufficiently often it was deemed worth the extra cost (1 byte!) of storing a copy. `SELF.PrimaryKey` is stored in the object for similar reasons.

Finally there is a loop filling in the nested queue with one entry for each key component. This queue has two gotchas to watch for. The first is it contains an `ANY`. You must always `CLEAR` a queue buffer containing `ANYs` before assigning to the fields (see the Language Reference Manual). The file driver can return the field number of a key component, and the `FileManager` has the record buffer in `SELF.Buffer` so we can use `WHAT` to return an `ANY` referring to the field of the key component. This then gets stored in the queue.

The other required piece of information is the "bind" name of the key component. This comes from `PROP:Label`, but the code does one more piece of pre-computation. If the key is case insensitive and the field is not numeric then we store `UPPER(Fieldname)` rather than `Fieldname` in the queue. This is neat as it means routines further down the line can simply use the stored field name without regard to case sensitivity and the issues are dealt with for them.

```
FileManager.Kill PROCEDURE
```

I suspect some would argue that `Kill` is not an initialisation method, but I disagree. `Kill` is really the mirror image of `Init`. I nearly always write them at the same time (literally a line of `Init`, then a line of `Kill`). The job of `Kill` is to undo anything that has been done during the `Init`, and sometimes it has to undo the work of later routines too, but acting as an `un-Init` is a must.

Notionally this routine is very straightforward but there is some quite involved picking apart of the queue of queues. The outer loop steps through each key and the inner loop steps through each key component. Finally for each key component the name is disposed and the `ANY` variable is freed up.

Failure to do any of these steps would result in a memory leak.

`FileManager.SetThread` PROCEDURE

As suggested in the [Static Usage](#) section one of the problems the `FileManager` has to contend with is that a file effectively has a new instance on each thread. This needs mirroring in the `FileManager`. Rather than clone the `FileManager` for each thread (and support an inter-communication layer) it was decided to support a queue within the `FileManager` where each thread was mirrored by a record in the queue. This may not please the object purists but as the thread specific data totalled 10 bytes compared to >1K for the whole object, the engineers will understand the logic.

The methodology is thus quite simple: each procedure starts by calling `SetThread` which pages in the correct data for the current thread; the procedure can then simply access the data using `SELF.Info`. There is a slight gotcha here. Because `SELF.Info` is a queue buffer, any method altering a value in `SELF.Info` must then do a corresponding `PUT(SELF.Info)` or the value altered will be lost upon the next call to `SetThread`.

The implementation is straightforward. If the file is threaded (early ABCs ignored this condition!) then the current thread is read from the system. The current list of thread data is looked up, and if it doesn't exist then a new record is cleared and created.

The final line of this procedure is actually a cheat. It has nothing to do with setting the thread; it simply takes advantage of the fact that all `FileManager` methods will call `SetThread` before commencing execution. `SetThread` thus because a suitable repository for always call items. In this case the error manager is being primed with the current file name so that any errors within the methods themselves can simply `Throw`, confident that pertinent details have been filled in ahead of time.

Summary

I hope this overview has been of some help. The `FileManagers` are coded in a fairly unusual context which has greatly influenced some of the design decisions taken during their implementation. In my next article I will go into the remainder of the methods looking at their code and design ideas.

[David Bayliss](#) is a Software Development Manager for Topspeed Corporation. He is also Topspeed's compiler writer and the chief architect of the Application Builder Classes.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

The Clarion Challenge

A String Parser

So you think you can write some tight Clarion code?

Try your hand at this problem. The challenge is to come up with the fastest Clarion code to meet the following requirement:

The Requirement

Write some code to parse a string. You might find this easiest to implement as a class, given the requirements.

A string parser can be a very useful part of your toolset. You might have to convert a list of full names to individual fields such as FirstName, MiddleName, and LastName, or you might want to format your programmers' source code in a consistent way, enforcing standards for Clarion keywords. Or maybe you just want to capitalize all words in a sentence. A string parser which reduces a text string to its components makes all these tasks easier.

First of all, you'll need to read in a string. If you're writing a string parsing class you may want to do this via an `Init()` method call.

Any string you parse will be made up of delimiters and tokens. Delimiters indicate where a token ends (i.e. in a sentence whitespace and punctuation marks are delimiters between words, which are the tokens). You must be able to add or remove delimiters at runtime so that the parser can be set up on the fly to deal with different kinds of strings. Delimiters may be longer than a single character.

You must also be able to retrieve the number of tokens in the string and any individual token. You must also be able to update a token and retrieve the entire string recreated with the updated token(s). Code to do this might look something like the following (this code fragment is not part of the requirement but is shown as an illustration of the requirement – you are free to implement your solution as you see fit):

```

LOOP X = 1 TO StringObj.GetTokenCount()
  TempString = StringObj.GetToken(X)
  ! do something with the string
  StringObj.PutToken(X,TempString)
END
Result = StringObj.GetString()

```

In the previous Clarion Challenge it became clear that compactness of code means almost nothing to the speed of the code. And in this challenge, speed is everything. So don't be concerned about how much code you're writing. Just make it fast.

Send your answers to advisor@clarionmag.com

The most compact and the fastest examples will be [posted here](#), along with any others deemed of special interest.



Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Clarion News

May 25, 1999

[ABC DLL Toolkit Beta For C5A](#)

Jim Katz now has a beta version of the ABC DLL Toolkit available for C5A. Any Toolkit users who would like to test the beta please contact Jim at jimkatz@ix.netcom.com.

[WIN API Examples](#)

Mark Geisinger has provided an example of using WNetGetUniversalName() to get the UNC name from a mapped path. This is a C5a ABC app. with a single source proc. Look for it in the Win API section at <http://hobbyist.bugsbugme.com>.

[Nice Touch Special Offer Expires May 31](#)

Two special offer prices will be expiring on May 31st, 1999. The View Wizard 1.0 introductory price will go to \$229 from the current \$199, and the NTS WizPak bundle price of \$649 (\$995 when purchased separately) will expire.

[Send Internet Mail 3.03 Released](#)

The Final version of Send Internet Mail 3.03 has been released, and is a free upgrade for all users of 3.x versions. This release features a choice of 16 and 32 bit demos and a number of bug fixes.

[C5 Accounting Modules](#)

KV Enterprises, Inc. currently has finished a complete Wholesale Distributor Accounting Package coded in C5. This royalty-free package is available with source code and consists of Accounts Payable, Accounts Receivable, General Ledger, Inventory Control, Purchasing, Order Entry, and Payroll. Work Orders and Bill of Materials are coming soon. A module for Fuel/Oil Jobbers is also available. Price is \$399.00 per module.

[Windows 2000 Beta 3](#)

Join the Microsoft® Windows® 2000 Corporate Preview Program (CPP) to start learning now about Windows 2000 Professional and Windows 2000 Server. MS says there is a limited supply.

[ABC Free Templates Help](#)

A first draft of the ABC Free Templates and Tools set is now available. Suggestions are welcome.

[Developers Wanted](#)

Topspeed Recruiting is looking for developers in the following states: California (4) , Florida (5) Miami (C.I.O & bi lingual developers), Texas (3) , Michigan (4) , Iowa, North Carolina, Connecticut, Arizona(2),and Colorado. The positions vary but all offer competitive salaries, full benefits and relocation.

[Oracle and AS/400 Accelerators Now Available.](#)

The updates for the Oracle and AS/400 Accelerators for Clarion 5A are now available as a free download. The updates must be applied to use the Oracle and/or AS/400 Accelerators with Clarion 5A. The Oracle Accelerator update requires both a prior version of the Oracle Accelerator

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

If you're interested
take our poll &
let us know!

for Clarion 5 and an edition of Clarion 5A. Likewise, the AS/400 Accelerator update requires both a prior version of the AS/400 Accelerator for Clarion 5 and an edition of Clarion 5A.

[C5B Candidate Release Available](#)

The Clarion 5 B Update for Enterprise Edition (Candidate Release 1) is now available as a free download. This patch program installs Candidate Release 1 of Clarion 5 Enterprise Edition release "B". This is an upgrade installation, and as such you must have Clarion 5 A Enterprise Edition installed (either installed from CD or updated with the patch).

[Clarion Text Extractor](#)

Dalby Data has released version 1 of its text extractor utility which extracts strings from a Clarion project for translation. Features include column text handling, multiline text strings, and filtering of non-translatable strings.

[Special Agent Released](#)

Special Agent is a new product from CapeSoft which builds the Microsoft Agent OCX directly into your 32 bit Clarion 5 program. Agent is an animated character that is commonly used as part of the help system (i.e. the MS paperclip). Special Agent features easy recording of demos and tutorials. Clarion 4 & 16 bit support is expected soon.

May 17, 1999

[NextAge's Pete Halsted To Speak At Jersey DevCovey](#)

Pete Halsted will be speaking at the Clarion Jersey DevCovey on the subject of OCX/OLE implementation in C5EE. His presentation will include samples of working with the Imaging OCX's, Microsoft Word, and Excel. The conference will be held at the Holiday Inn at Princeton, NJ on May 22, 1999. A copy of NextAge's imaging template will be one of the door prizes.

[New Handy Tools Build](#)

There is a new Handy Tools build available, incorporating some minor fixes. Features recently added to Handy Tools include a query parser to enable complex filter strings, and Windows-like record marking in browses. As well, Handy Tools can now be registered online. The price for the unlimited version is \$99.

[G-Calc 4.2 Now Available](#)

Gitano Software has released G-Calc version 4.2. New features include new path/INI options, wallpaper settings, paste option, and better support for hand coding. Version 4.2 is free to registered users of version 3.0 or higher.

[Clarion 5A Web Edition and IC Updates](#)

The Clarion 5 A Web Edition is now available as a free download, as is Internet Connect for Clarion 5a. Use this patch if you have Enterprise or Professional Edition plus Internet Connect.

[Catalyst SocketTools Examples](#)

Mark Geisinger has set up a web site in response to interest in his Catalyst SocketTools Library Edition examples. The site includes the following: FTP - performs asynchronous FTP download; SMTP - email delivery with optional attachment; Sockets - simple Echo session using TCP sockets.

[More Catalyst Tools!](#)

Vince Sorenson has made a first draft of some FTP wrapper classes for the Catalyst Library Edition FTP DLL available in his ABC Free Templates and Tools set. The first draft can be downloaded from .

May 10, 1999

[Imaging Templates Updated](#)

Version 1.03 of the Imaging Templates has been posted. This version fixes a bug that would cause a compile error when using BLOB's to store the images files. Also note that the imaging OCX's will only work with a

32bit program.

[SocketTools ABC Example Updated](#)

The C5a ABC example of using the Catalyst SocketTools Library Edition to perform SMTP email transfer has been updated to include sending an attached file. Be sure to read the readme, especially if you decide to build the program.

[Legacy App Migration Handbook Planned](#)

Bob Foreman of TopSpeed is looking for feedback on a planned handbook for migrating apps from legacy to ABC templates. Possible topics include conversion tools, learning ABC, and chapters on ABC techniques for general application development, data and file access, windows and controls, browses, lists, and trees, and reports and processes. [Email Bob](#) with your suggestions.

[Sterling Data Announces Searching/Tagging Template](#)

Sterling Data has released SearchFlash, a template which provides a range of searching/tagging options for Clarion browses. Features include text field and/or QBE searches, child file searches, automatically displaying the matching field/record, tagging button controls, and templates for processes and reports. Supports ABC and legacy. Source code is included. List price: \$95.

May 3, 1999

[IVR Template In The Works](#)

Carl de Billy is presently making an IVR (Interactive Voice Response) Template for Clarion and would like to know if anyone is interested in this product. It will support Dialogic, Rhetorex (Lucent) boards and some voice modems.

[PeaBrain Looking For Feedback on DM 5](#)

Duppy Du Preez has added a Logical view option to DM 5, but needs some ideas on how to make this really work for Clarion developers. This is not an Entity view (showing a block connected to a block) but should be a view that points out your logic behind the design. [Email Duppy](#) for more information or see message 11779 in Products.C5EE on the TS news server (tsnews.clarion.com).

[C5A Released!](#)

Clarion 5a for Enterprise Edition and Professional Edition are now available as free downloads. Enterprise Edition updates are available from www.topspeed.com/c5ee-upd.htm. Professional Edition updates are available from www.topspeed.com/c5pe-upd.htm

[Gitano Easter Egg Hunt On Again](#)

As no one found the easter egg in GCSetup last time, the hunt is back on. Find the egg and win a license to GCal Pro or GCal/GCalc/GNotes bundle or GReg Plus.

[SMTP Mail Transfer Example](#)

Mark Geisinger has provided an example program which uses the Catalyst SocketTools Library Edition to perform SMTP mail transfer.

[Read the April 1999 News](#)

Do you have a news story or press release we should know about? Send it to editor@clarionmag.com

Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Feature Article

Demystifying The Debugger

by Russell B. Eggen

In some of my recent classes and a few newsgroup articles, I was asked to explain or show the debugger. Those in my class may testify that they could have seen a wicked looking smile. Those asking on the newsgroups let me assure you that the smile was indeed there.

I really don't know why the debugger has fallen out of use so quickly. I am big fan of using debuggers.

In CPD 2.x, Clarion users had a debugger, sort of. Remember all those CTRL-S and CTRL-F keystrokes to see one line of code? Not too many things to inspect, but it did work, slow as it was.

In version 3.x, Clarion Software (Topspeed's former name) delivered a really robust, state of the art debugger. I loved it! At the time I was also working in a COBOL shop. Micro Focus made one of the best debuggers I have ever seen. Reading pages of COBOL code is quite a task, and to navigate through so much very quickly is quite a task. The CFD debugger had about the same power and ease of use. No wonder this was one of my most favorite versions of Clarion. (OK, I am a minority on this point, but that is my story and I am sticking to it).

When Clarion for Windows first came out, it wasn't long before I hit the debugger. What is an event anyway? Why did I have to go through the ACCEPT statement so many times and why did I have to step through all those lines of code before I got to the interesting bits?

The debugger seemed OK, but why did I have step through all those routines? I was not interested in those, just my code. I gave up just like everyone else and started using the MESSAGE() function. But that takes some typing to set it up right. So I began using the STOP() statement.

When I couldn't figure out why some code was behaving the way it was, or it started working with all the STOP() statements in place, I was even more mystified! One of my co-workers suggested that I use the debugger. Oh NO! Not the dreaded debugger! PLEASE! I was in a hurry, deadlines approaching. I felt like I was fighting for my life. I fired up the debugger and stepped through every line of code in my procedure. I did eventually find the problem and fixed it. But what a chore that was!

A couple of years ago I had the fortune of talking with designer of the debugger. I started asking some questions. He answered them. I still remember the last question I asked him: "How long has that been in there?!" After we finished our drinks, I had to go back to my room and try a few things. Son of gun! Will ya look at that! I was getting excited about debuggers again! There were the breakpoints, the fast stepping, and even watch windows! He told me how these features worked.

However I wasn't completely convinced. At the time I was having problems with a conversion program, and I had STOP() statements everywhere. The problem was all the values shown by my STOP() statements were correct, but the application still wouldn't work properly.

BKO
Enterprises, Inc.

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

etc
2000

If you're interested
take our poll &
let us know!

I was sure there was nothing wrong with my code.

Out of desperation, I fired up the debugger, and I found the bug in 20 seconds. This after four hours of reading the code, `STOP()` statements, etc. The problem wasn't that my code had bugs. In fact, the debugger proved that my code was good, but it did not work! The code I wrote did not have bugs in it; the code I did not write was the problem.

Huh? The problem was that I forgot to add code to `OPEN()` a file. My bug was missing code! How can `STOP()` or `MESSAGE()` ever find that problem? Of course this exposed a few more lines of code I needed to write (error checking, adding or changing the converted data, etc). Needless to say, it was not one of my better moments (or programs)!

In this case, the debugger showed to me that there was an error being returned by the `PUT()` statement: "File not open." `MESSAGE()` and `STOP()` can report this, but you have to write the code to display the error. If I had thought of that, I would have written the error checking in the first place! This lesson taught me a few things, and the first one was humility. In this example, `STOP()` and `MESSAGE()` would not have helped. Neither will be of much help in getting into deeper issues either, so you might as well stop using them.

Why not use `STOP()` and `MESSAGE()` for debugging? Well, let me be a bit blunt. Neither is designed for debugging! `STOP()` is designed to suspend program execution (see the Help topic). This means it requires a response from the user. The `MESSAGE()` procedure displays a Windows-standard message box, typically requiring only a Yes or No response, or no specific response at all. It is also interacting with a user.

These functions process events too! Have you ever had a procedure filled with `STOP()` or `MESSAGE()` statements, and it started behaving strangely or went on an entirely new and wrong path? Guess what was messing with your event messages? What about all those embeds you put these functions in, only to get a phone call from your customer that he got a `STOP()` message! What guarantee is there that you removed all of them before you shipped the product?

I hate to break this to you, but `STOP()` and `MESSAGE()` are not safe debugging tools! You have enough to debug as it is!

Enter The Debugger

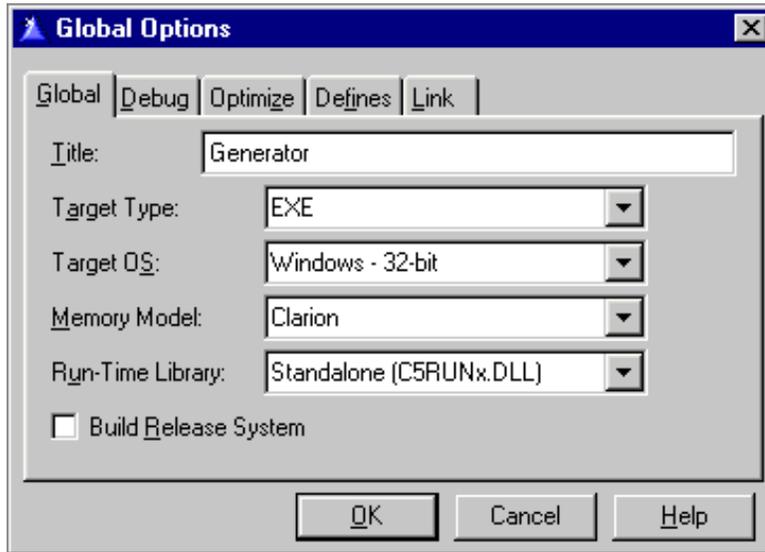
OK, I admit it. I prefer the 32-bit debugger to the 16-bit. I have my reasons. I think the interface is cleaner and easier to navigate quickly. Also, the startup process makes it easier to get to the code where I think the bug lives. You can also use the 32-bit debugger on NT. The 16-bit debugger won't run on NT - period. NT is designed to crash low level 16-bit applications. And I also like how much easier it is to add watch variables in the 32-bit debugger.

I will now walk through the debugger using the MailList application, which you can find in your Clarion /examples directory. First I'll explain typical debugging techniques, and later I'll show you how the debugger is also a good testing tool (this is one of the secrets I discovered that does not require anyone to pick up a dinner tab). Also I'll give you some tips for using the debugger outside of the Clarion IDE (yes, you read that correctly).

The Basics

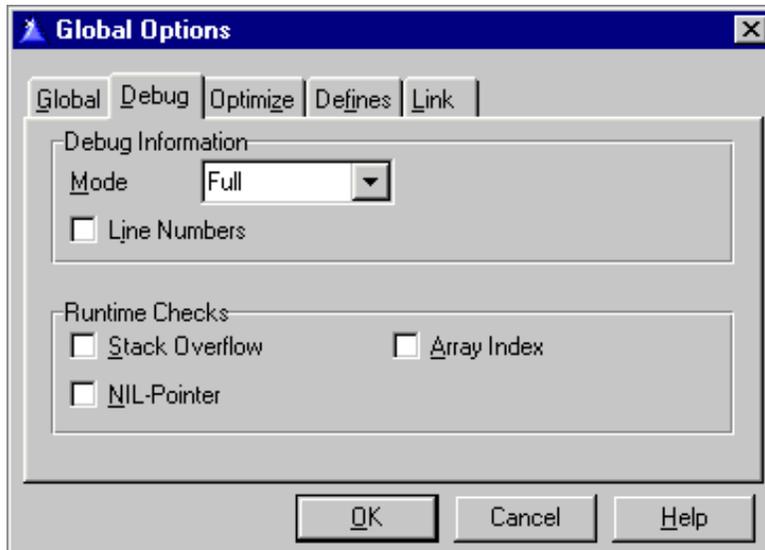
Any project you want to use the debugger on must have Debug turned on in the project settings (this is the shipping default). See Figures 1 and 2. If you have changed these settings to Build Release then you must uncheck this and set Debug to Full. You'll also need to set the Target OS to Windows - 32 bit, as this article focuses on the 32 bit debugger.

Figure 1. The global project dialog.



TIP: If you use string slicing or other code to step through arrays, then set the Array Index checkbox on. This will trap out of range subscript errors..

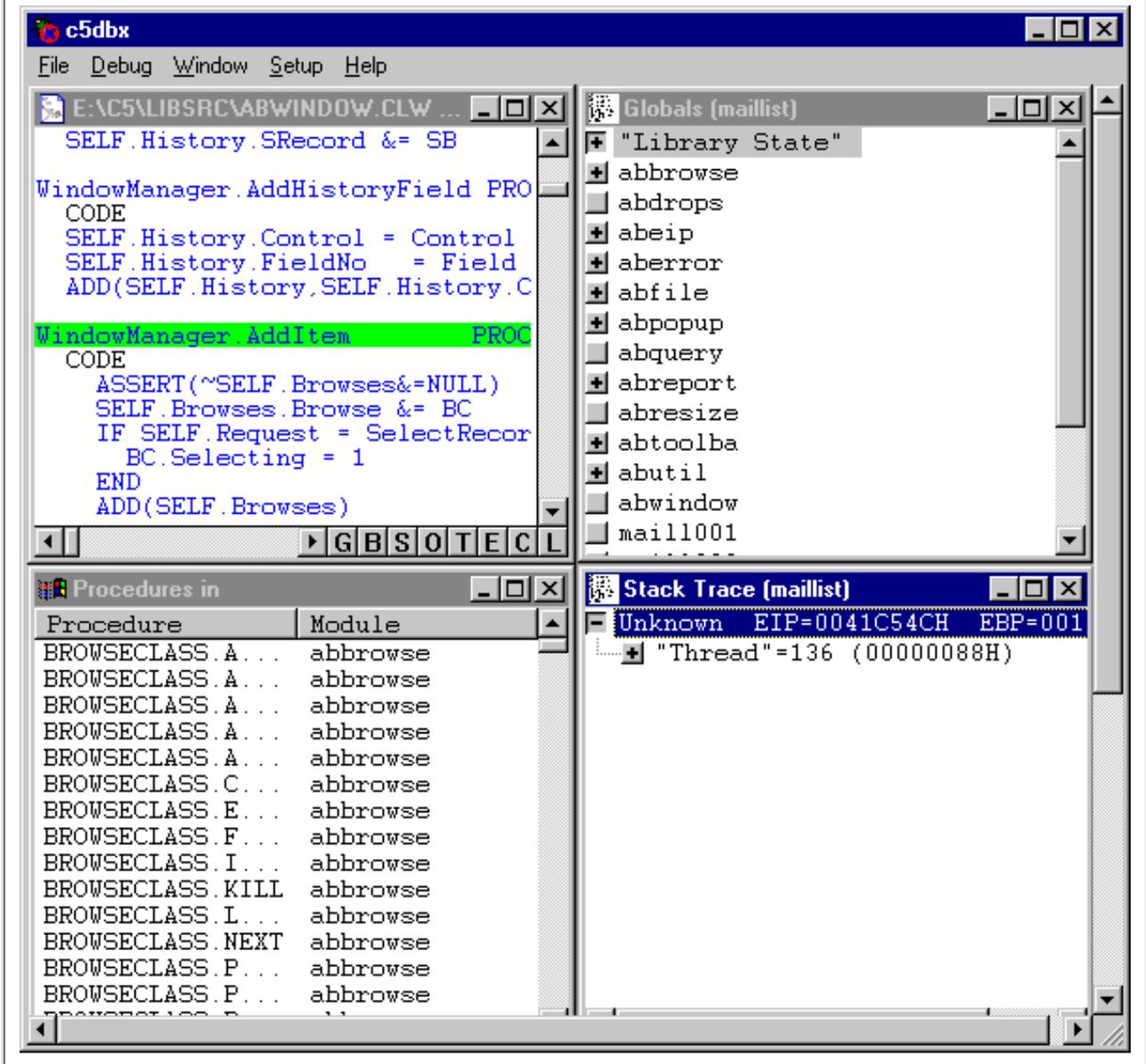
Figure 2. Global debug settings with array checking on.



Save your project settings and choose Project|Debug from the main menu or click on the debug button on the toolbar. (Editor's note: If you still have some Clarion 1.0 beta disks sitting around in your archives, dust them off and have a look at the debugger EXE with a resource editor – you'll see some entertaining alternative images to the standard bug-with-the-red-line-drawn-over-it)

After compiling and making the executable, you see something similar to Figure 3:

Figure 3. The 32 bit debugger at startup.



Now what? The first thing I do is select the procedure I want to debug. You can select more than one, but let's keep it simple for now. I will debug the `BrowsePeople` procedure. Scroll down the procedures list until you find that procedure name.

TIP: The procedure and module columns are sortable. The default sort is the procedure column and it uses a step locator. Click on the heading for module and the list resorts accordingly.

At this point, as soon as you click on the procedure name, its source file is instantly visible in the source window. I minimized the rest of the windows at this point and resize the source window so that it fills about 3/4 of the visible screen (it can go virtual if you stretch it past the dimensions of the screen, but I don't like that, so I don't use it).

At the bottom of the debugger window, you have the Trace, Disassembler, Procedure (See figure 3), Stack Trace and Globals windows, all minimized. Here's a brief tour of some of these windows.

What if you want to see a variable defined in your dictionary? Since variables declared here are considered global, restore the Globals window. Where are the file variables? The entire display is collapsed (it is a tree display). All dictionary variables are under the node named after

your application, in this case, MailList. Variables declared under the global button in your app can also be found here.

Expand the People tree. All variables are sorted alphabetically to make them easier to find. Highlight `PEO:FirstName` and right click- a pop-up menu appears. There is only one item on the pop-up: Watch Variable. Click on this and presto! The watch window appears. It is at this time that I like to resize and move the watch window under my source window. Now add `PEO:LastName` in the same fashion. You should have two variables in the watch window. Since the debugger has not yet begun to move through the code, the contents of these should be blank.

If I were to add numeric data or single elements of a string (strings are arrays of characters), then I would have had two more items in the pop-up window, Edit Variable and Examine Memory. You cannot edit an entire string, but you can edit or examine a specific string array element. Expand the `PEO:LastName` variable and then right click on one of the elements to see these options.

Minimize the global window as there is no further need of it. Move and resize the two open windows so they are orderly. Here is how my screen looks like after I have completed this step:

Figure 4. Source code in the debugger.

```

MEMBER('Maillist.clw')                                ! This is a MEMBE

INCLUDE('ABBROWSE.INC'), ONCE
INCLUDE('ABPOPUP.INC'), ONCE
INCLUDE('ABREPORT.INC'), ONCE
INCLUDE('ABRESIZE.INC'), ONCE
INCLUDE('ABTOOLBA.INC'), ONCE
INCLUDE('ABWINDOW.INC'), ONCE

MAP
    INCLUDE('MAILL002.INC'), ONCE                    !Local modul
    INCLUDE('MAILL001.INC'), ONCE                    !Req'd for m
    INCLUDE('MAILL003.INC'), ONCE                    !Req'd for m
END

BrowseMailers PROCEDURE                                !Generated from p
CurrentTab      STRING(80)
PeopleText     STRING(40)
PeopleCount    USHORT
LocalRequest   LONG
FilesOpened    BYTE
MailerText     STRING(30)
BRW1::View:Browse VIEW(Mailers)
                PROJECT(MAI:Description)
                PROJECT(MAI:MailerNotes)
                PROJECT(MAI:MailerNumber)
                END
Queue:Browse:1 QUEUE                                  !Queue declaratio
MAI:Description LIKE(MAI:Description)                 !List box control
MAI:MailerNotes STRING(SIZE(MAI:MailerNotes))        !Browse hot field
  
```

Stack Trac... E:\C5\LIBS... Procedures... TRACE Disassembl... Globals (ma... G B S O T E C L

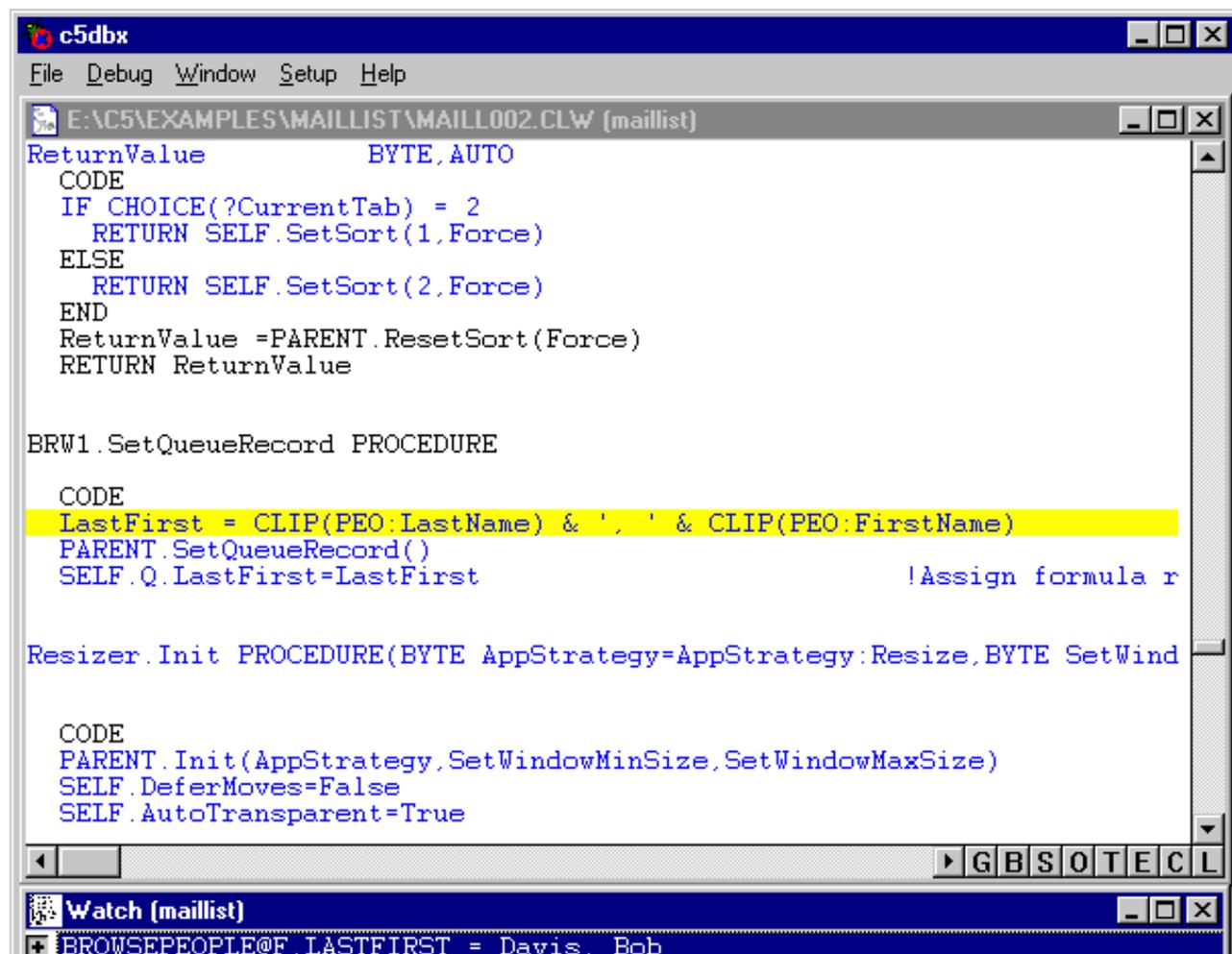
Where Should You Debug?

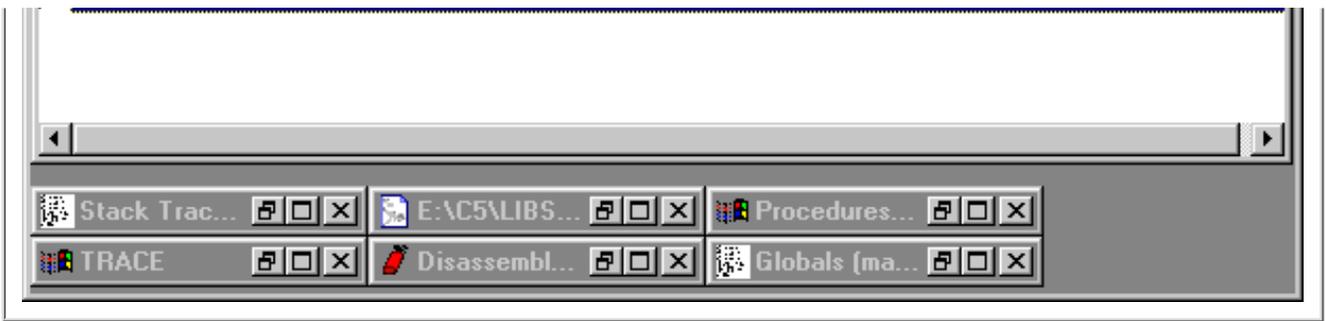
Page down to `BRW1.SetQueueRecord` while in the source window. Click on the first line after `CODE`. This line formats the name fields into a local variable. The line should be green in color. Press the letter "B" or double click or right click and select `Set Breakpoint`. This sets a hard break. Hard break means that until the debugger hits this line of code, you do not have to step through each line until you get to this point. The debugger gets focus when this line of code is hit. The line also changes color to red indicating a hard break point. It isn't red? Press the down arrow key to highlight the next line. It is now. The current position is green and hard breaks are red. Red and green make yellow. Cute, eh?

Now you are ready to debug, sort of. You cannot add any local variables to the watch window as you are not really in this procedure. But you soon will. To start the debug process, I like pressing the letter G (or you can right-click and choose `Go` from the popup menu, or from the pulldown menu, `Debug/Go`). Your program is running normally.

Now choose `Browse|People` to send mailings. The debugger now gets focus at the breakpoint. Since Clarion uses a stack calling convention (a fancy term for resource management), you will see that there is also a `Stack Trace` window. This is where you will find your local variables as they have now been pushed onto the stack. Restore the stack trace window. Expand `BrowsePeople`. Now find the local variable named `LastFirst`. Right click on this and choose `Watch Variable` to add it to the watch window. You see the variable is now blank. Minimize the `Stack Trace` window to move it from the workspace as it is no longer needed. The source window should be the active window.

Figure 5. Source code and watch variables in the debugger.





If you right click and choose Step Source (or press the letter T), the current line moves down one line of code and right before your eyes, `LastFirst` is changed in the watch window.

Now for some fun press G and wait for the debugger to go around and update the global variables. Now press T and watch the local variable change to its new contents. It will do this until the list box is filled.

You can set as many break points as you wish to get the job done, inspecting the contents of any variable you wish (as long as it is in scope). Remove the hard break, by repeating how you set it. Now press the Go button and your app runs normally. You can now choose another source file to debug, if you wish.

If you are done debugging, there is something very important you must do. Close down the client application normally, then close down the debugger. Closing down the debugger with the debuggee still active could cause memory leaks, or worse.

What Did You Discover?

- You discovered that you do not have to step through every friggin' line of code. If this was required, I think I would take my chances with `STOP()` and `MESSAGE()`.
- Going right to a line of code you want to check out, setting breakpoints, and then running the application until a breakpoint is hit is much faster than coding `STOP()` and `MESSAGE()` statements (and safer!).
- You have no code to remove once you are done (and no danger of forgetting an embed used for debugging and having your customer find it).

What about debugging DLLs?

DLLs are applications too, and the same rules apply.

Here's a common example. Do you use a data DLL (no procedures, just `dict` declarations and template variables)? If this DLL's debug options are set to Release Mode, guess what you cannot see, let alone place, in a watch window? Global data. This means every field defined in your dictionary. You won't find a one. So remember to change your project settings there too, before you debug.

Tips and Tricks!

One of the tricks I have "taught" the debugger to do (or is it the other way around?) is to assist me testing an application. Any good developer tries to predict what a user can and will do and writes code accordingly. How many of you have had customers call you and say, "I did this and it went bang?" Customers are a very creative lot.

As an example I will use a program that simply reports the event that has fired. This application is used in the Essentials class, so any former Essential student has this program and can follow along. You can also [get the application here](#). The `NameEvent` routine that returns the event string also checks for a user defined event and unknown errors. The code looks sound (it is quite simple), but here is the question: How can I test this to prove that it works? I don't want to write code that force

things down a certain logic path, as that can be risky. Writing more code, no matter how simple, adds complexity to a project, increasing risk of logic errors. What would be ideal is that I steer the program down the "once in a blue moon" logic path, without writing any extra code. Besides, I am too lazy for that!

I can now safely test each condition coded to see if it works at runtime, just like your customers will do. Then I know it works as I have seen it perform as I designed it. Is that not how you should test an application, like a customer would use it?

What I need to do is setup my project to use the debugger (again 32-bit) and start it. In this case, all the code is in one source file. The code I am interested in is at the end of a function that returns a string informing the user what event has just executed, as seen in Listing 1.

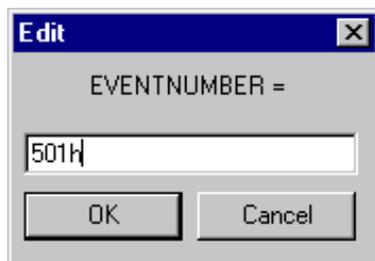
Listing 1. Partial source listing showing what the debugger will test.

```
NameEvent    FUNCTION(EventNumber)
ReturnString  STRING(20)
CODE
CASE EventNumber
OF 01H
    ReturnString = 'EVENT:Accepted'
    ! Many more event tests omitted
OF 213H
    ReturnString = 'EVENT:Maximize'
OF 214H
    ReturnString = 'EVENT:Minimize'
OF 215H
    ReturnString = 'EVENT:Completed'
OF 400H TO 0FFFH
    ReturnString = EventNumber & ' - User Defined EVENT'
ELSE
    ReturnString = EventNumber & ' - Unknown EVENT'
END
RETURN(ReturnString)
```

As you can see, this code is part of a large CASE structure, so I have set my break point at CASE EventNumber. At the first event that fires, the debugger gets focus because least one event has fired. Now I need to bring up the Stack Trace window and find the local variable called EventNumber (which is also seen in figure 5 above). Once I do this, right click and select Edit Variable.

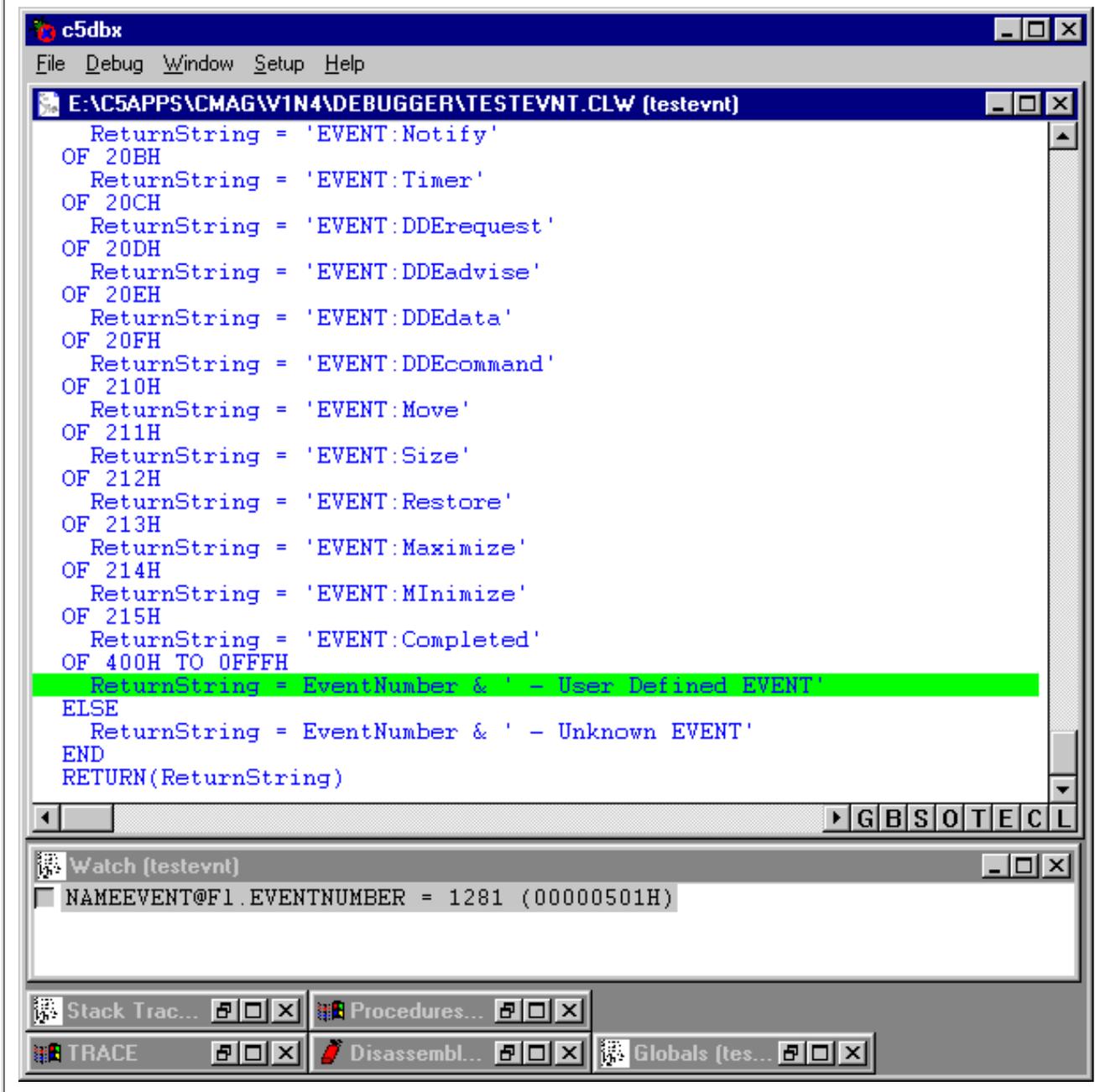
Here is something useful. The listing above is testing hex numbers, but the edit window is prompting for a decimal number. No problem. Simply enter the new value as shown in Figure 6:

Figure 6. Enter a hex number by typing 'h' after the number.



Click on OK and then "Step Source". Notice in Figure 7 where the cursor has landed?

Figure 7. Detecting a user-defined event.



The code works as expected and now I have proof it works. The value 501h is certainly between the range reserved for user defined events and the code is picking this up. I can also test for the out of range condition by giving some value that cannot be trapped in this case structure. In this case, I would get the "unknown event."

Moral Of The Story

The debugger can test for all kinds of conditions in your code. If the code doesn't work, then it is a simple matter of setting a few more breakpoints and clearing breakpoints no longer needed until you narrow down what the exact problem is.

A very good friend of mine once told me "the correct 'why' opens the door for handling." And my grandfather told me that all solutions are simple. The debugger is certainly living proof of both sayings.

Sometimes the biggest bugs in our code are staring at right at us, like my embarrassing example earlier. In my experience, the nastiest bugs are hiding in plain sight. It is a relief and a real shout of "doh!" or a slap

of the forehead when you finally find them. Once you find the problem, then writing the correction is quite simple.

Registry Settings

If you would like to run the debugger outside of the Clarion IDE (i.e. right click on your EXE and launch the debugger), then try adding these entries to the registry (backup your registry file first - you have the tools needed to do this already. Regedit will do this and Windows 98 will do this upon startup). You may need to change the Clarion drive and/or path to suit your own machine.

```
[HKEY_CLASSES_ROOT\exefile\shell\16bit Debugger]
[HKEY_CLASSES_ROOT\exefile\shell\16bit Debugger\command]
@="c5db.exe c:\\clarion5\\bin\\clarion5.red c:\\clarion5\\bin\c5ee.ini %1"

[HKEY_CLASSES_ROOT\exefile\shell\32bit Debugger]
[HKEY_CLASSES_ROOT\exefile\shell\32bit Debugger\command]
@="c5dbx.exe c:\\clarion5\\bin\\clarion5.red c:\\clarion5\\bin\c5ee.ini %1"

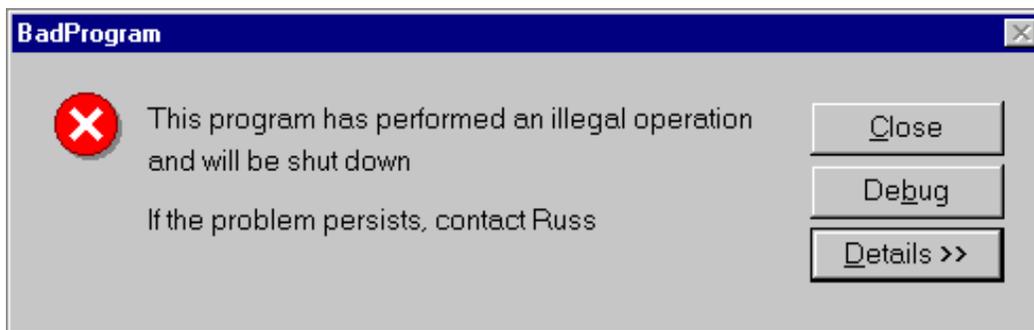
[HKEY_CLASSES_ROOT\exefile\shell\Restart Debugger]
[HKEY_CLASSES_ROOT\exefile\shell\Restart Debugger\command]
@="c5db.exe /r"
```

These settings will launch the debugger from Explorer with a right click (pop-up menu). The last grouping restarts the 16-bit debugger with all of your previous settings and windows. This is not yet available in the 32-bit debugger.

Post Mortem Debugging

If you install the 32-bit debugger as your system debugger (from the debugger's Setup menu), then when your program that crashes (GPFs), you get this familiar window with a small addition, as shown in Figure 8.

Figure 8. GPF with debug option.



If you press the debug button, then the 32-bit debugger starts and tries to find the source that caused the problem. This works with non-Clarion programs too (although you may not have the source), although in this case you will get the disassembly window.

OK, I have a confession to make. I have used the debugger with a bit of an evil twist.

As I stated above, this works on other programs. I have used this to report bugs to other vendors. I simply grab the screen shots of the disassembly window (which shows the code that crashed) along with the memory register contents. I then send the contents to the support folks and ask them "When will I get my fix?" I would be less than honest if I said that I did not get some sort of satisfaction with that statement! In case anyone is interested, I did get a fix! And it worked. If anyone is curious, it was a VB program – 'nuff said.

Conditional Break Points

The debugger does not support conditional breakpoints, I regret to say. This missing feature is known about and when it will make its debut is beyond the scope of this article. However, you are not stuck. You can code your own break points. How?

Imagine you are processing a large number of records (lets make it a cool half million). The process hangs or goes bang every time the process reaches record number 486,457, just when you're almost done! The problem is that you are using a key and filters and this means that physical record number 486,457 is very likely not the cause. Give the debugger some embedded code to check for. In `ThisProcess.TakeRecord`, you can write the safe code in Listing 2.

Listing 2. Code for the debugger to watch.

```
?IF Count# = 486457
?  i" = 'Set hard break here '
?ELSE
?  Count# += 1
?END
```

Since this code is for the debugger only, there's no problem with using implicit variables. It really is throw-away code. So set your hard break on the string and choose Go. When the condition is met (486,457th record in sequence), the debugger becomes active. Now go find which record this is (by adding variables to the watch window) and use tools such as TopScan to find out what is this record's problem. I have run into problems with programs I wrote only to find the source of the problem was bad data, so be on the lookout for it.

It's your turn to try this. How would you write conditional code to find out when a value changes? Sometimes you need to test for a condition that should not change, but it does.

This technique takes almost no overhead, and if you leave it in your code it really doesn't do any harm. The trick is to keep debug code totally encapsulated. It is far more dangerous to use real variables as other processes can come along and change the values.

If you really do not like using implicits (and there are a few of you out there), then add variables in a suitable data embed. That is equally as safe.

TIP: The "?" symbol means that this code is compiled only when debug is turned on. If you switch to release mode, then the code is not compiled at all.

Summary

I certainly hope that you got some good information out of this article. You should now see that the Clarion debugger is a more useful tool than most people know.

I showed you how to step through code that you are interested in. This means that you do not have to step through all manner of code that has nothing to do with your problem, just the lines that you need to look at.

You can use the debugger to test logic flows in your code (especially those that may not be run that often).

I showed some setup tips that make getting ready to use the debugger easier to use. I also explained why `STOP()` and `MESSAGE()` are not really designed for debugging and why they are not safe to do so.

I enjoyed writing up my notes and sharing these with you. I am still wearing that smile. In all of these discussions, I have to ask you one last question:

Did anyone see me debug the ABC classes?

[Russ Eggen](#) has been using Clarion since 1986. Before joining TopSpeed as a consultant in 1996, he was an independent contractor. Currently, Russ is an instructor at TopSpeed and is writing the curriculum for the classes. His main goal in life is to get a Clarion program to star in a Tom Clancy movie where the program helps the hero save the world.

Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

The Clarion Advisor

Fast ASCII Files

ASCII files aren't often used for storing application data for several reasons: they don't have any native support for indexing, you can't change the length of an individual record without rewriting the file, only `STRING` and `GROUP` data types are supported, and some functions like `PREVIOUS` aren't available. As well, anyone with a text editor can alter the contents of the file.

Still, ASCII files have their uses, particularly for importing/exporting data and analysing text files. If you're working a lot with ASCII files there are some tricks you can use to improve performance.

The ASCII driver reads in data one buffer at a time rather than one record at a time. You can specify the buffer size with the `FILEDRIVERS` buffer string (see `Driver Strings` in the Clarion help file) where each buffer allocated takes 512 bytes. If you use a driver string of

```
/FILEBUFFERS=64
```

you'll be reading in the data in 32k chunks which should give fairly good performance. The larger the buffer, the fewer disk reads. However, to avoid possible record contention the driver will reread the buffer before each record access unless you turn `QUICKSCAN` (another driver string) on. Another approach is to open the file denying others write access.

A more low-level approach is to use the DOS driver instead of the ASCII driver. The ASCII driver looks for records that end (by default) in `CRLF`, whereas the DOS driver simply reads in the amount of data you tell it to read. You could read in data in 32k chunks and break it down into records by looking for the `CRLF` (or whatever record terminator the file uses). If you do this keep in mind that the record terminator will almost never be at the end of your 32k chunk of data, and there may be any number of terminators within that 32k. You could even have one half of the terminator in the end of one block, and the second in the start of the next block. As well, the last chunk of data will almost certainly be less than 32k in length.

If you're willing to go right to the metal then you can use the Windows API to read the file. Allocate a large buffer with `GlobalAlloc` (say one megabyte) then read data into the buffer with `CreateFile` and `ReadFile` in 32 bit, or `_lopen` and `_hread` in 16 bit.

Finally, if all of this API magic has you reeling, you could always wait for the next release. The C5B ASCII driver has been tweaked, resulting in a significant increase in performance.

Special thanks to Jim Kane, Arnor Baldvinsson, Gordon Smith, and Scott Ferrett for their suggestions.



Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton



If you're interested
take our poll &
let us know!

Copyright © 1999 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.



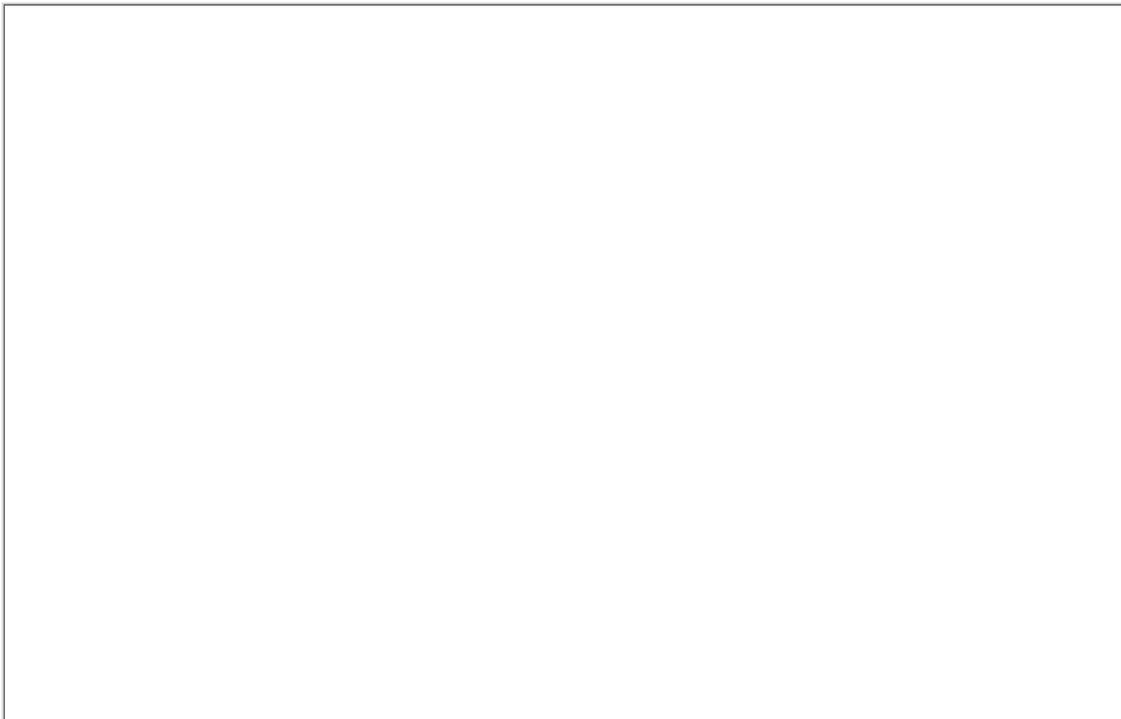
Clarion magazine

- Main Page
- Log In
- Subscribe
- Open Source
- Links
- Mailing Lists
- Advertising
- Submissions
- Contact Us
- Site Index
- ClarionMag FAQ
- Download PDFs
- Search ClarionMag

Feature Article

Clarion Developers Conference Latin America

The second edition of the Clarion Developers Conference Latin America was held May 13-15 in Buenos Aires, Argentina. Russ Eggen of TopSpeed was among the attenders, and supplied Clarion Magazine with the following photos. Thanks, Russ!

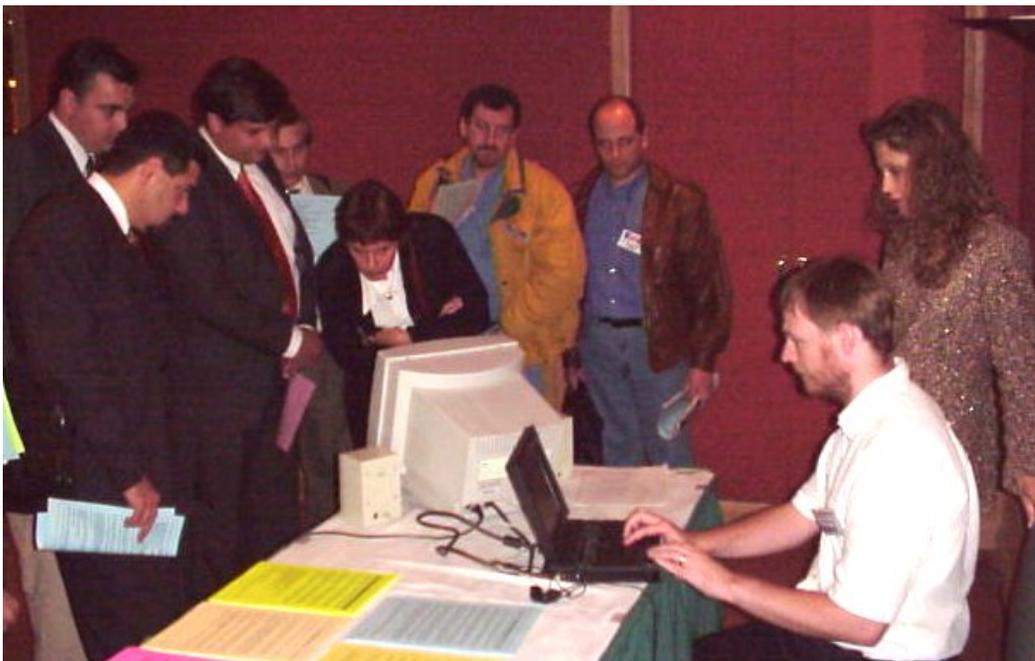




Kathleen Eggen, Cliff and Debbie Court, and Sue Rafalco.



Attendees listen to translated lecture (or is this the UN?).



Bruce Johnson showing CapeSoft products. The translator is standing behind him.

Adrian Brcic (Director General Unisoft)
and Antonio Rajan



After the first day of
the conference, a



celebration dinner is in order.



Gordon Smith in action the next day.



The start of day three - out in the countryside for breakfast.

Those are real polo ponies!





Breakfast over, heading over to the last conference.

The ranch owner's hacienda.



Copyright © 1999 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.



Clarion magazine

Main Page
Log In
Subscribe
Open Source
Links
Mailing Lists
Advertising
Submissions
Contact Us
Site Index
ClarionMag FAQ
Download PDFs
Search ClarionMag

Technology Preview

Beta Preview Linder Software's SFX Setup Builder

by Dave Harms

NOTE: This is a technology preview, not a full review.
Product features are subject to change.

There's been a lot of discussion recently (not all of it pleasant) about the Wise installer included with Clarion 5 Enterprise Edition. A number of developers have pointed out other options, and one that deserves a close look is Linder Software's SFX Setup Builder version 3, currently in beta 4.

This new version of Setup Builder is a significant upgrade, and incorporates many features including (but not limited to) high compression, password encryption, multi-volume installs, INI file and registry support, OCX/DLL/EXE registration, 16 and 32 bit installs, binary patch creation, and for 3rd party developers, Clarion version detection and automatic registering of Clarion templates.

There are two key concepts in the SFX approach to designing install packages: components and setup types. You use components to create groups of installation files which can then be treated as a single entity. For instance, if you're distributing a Clarion app with the runtime DLLs you may want to make those DLLs a separate component from the main app, and make them optional if you're selling to people who may already have Clarion (or one of your other applications).

All installs by default start with one component, and you can create as many additional components as you like. You can also specify whether an individual component is required or optional and where that component should be installed.

Setup types let you create installation options for the user, such as Typical, Minimum (for laptops), and Custom, one for each combination of components the user could wish to install. You can also specify which of the components are required and which are optional. All of this is done in a point-and-click interface. You don't need to write any scripts.

To test Setup Builder I created an install package for the open source debugging and profiling classes. There are only five files in this package: two class headers, two class source files, and one template. I made the class source one component, and the template another.

The main window of Setup Builder has tabs selecting files, setting options, INI/Registry settings, program icons, and programs to start after the install is complete. Figure 1 shows the file selection tab.

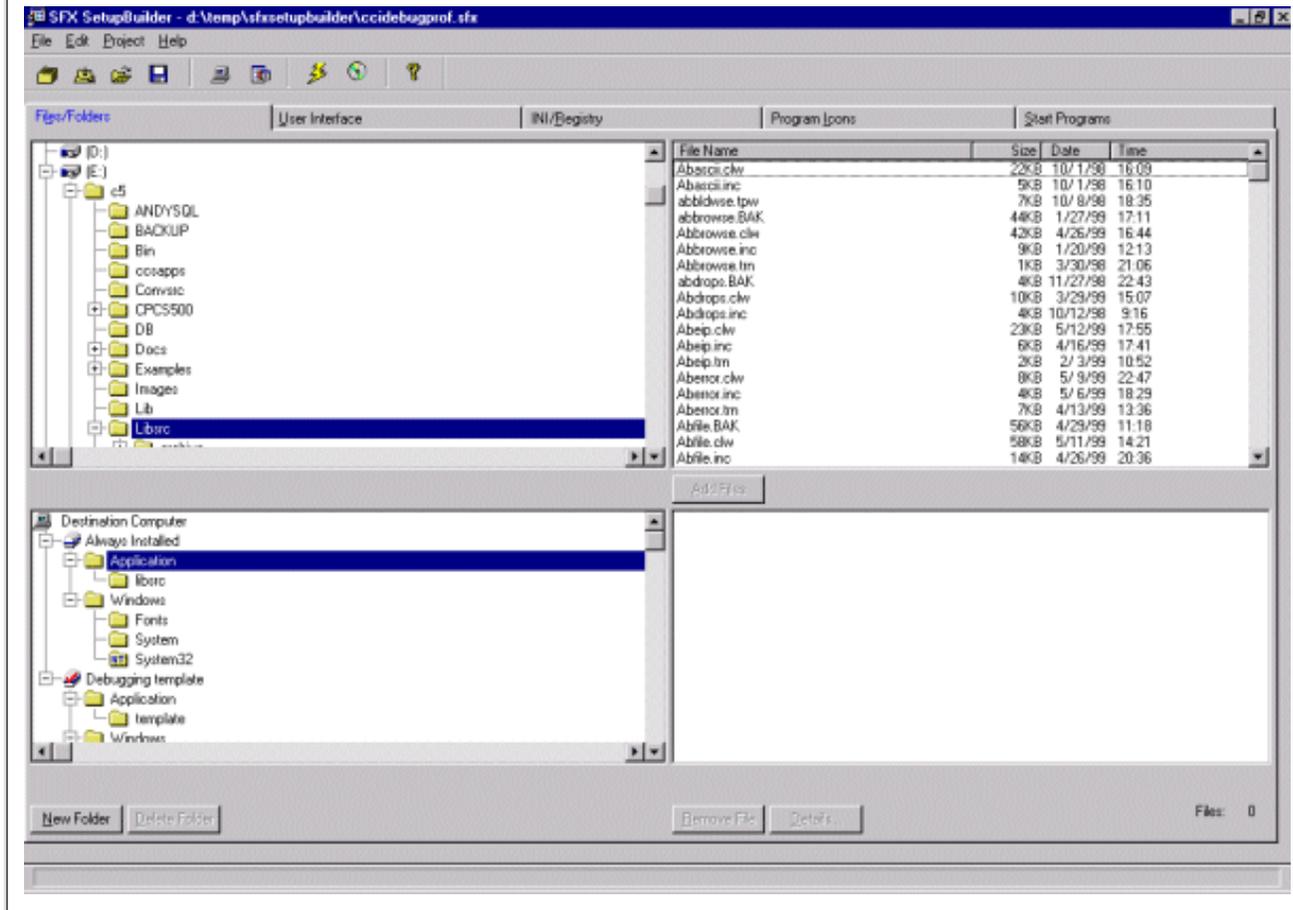
BKO
Enterprises, Inc.

Great Opportunity!
Salary to \$125,000+
Pre-IPO Stock Opts
Sunny Boca Raton

etc
2000

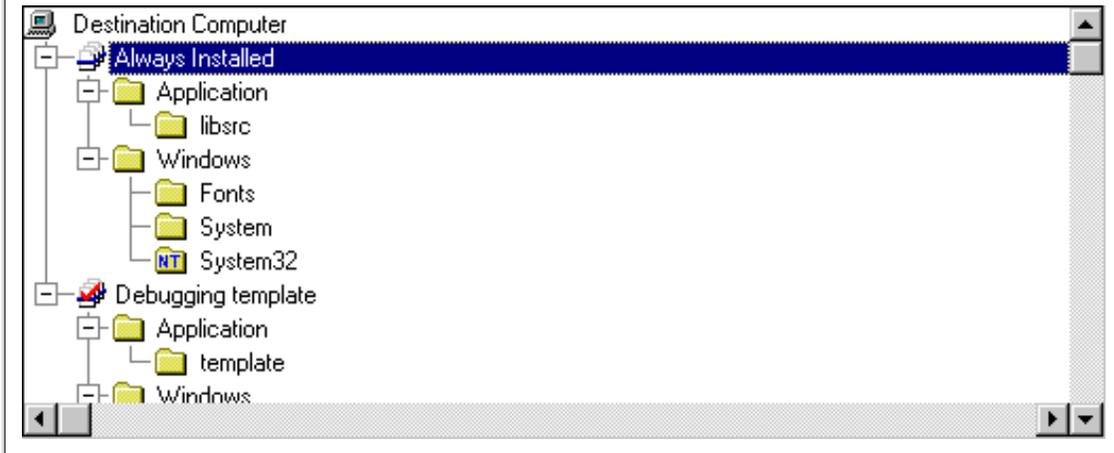
If you're interested
take our poll &
let us know!

Figure 1. The Files/Folders tab.



The upper list boxes let you select the files you want to install. The lower list boxes show you the directories on the destination computer, and let you specify where you want particular files placed. Since I wanted to install the class files into the libsrc directory and the template into the template directory, I first had to create those folders, as shown in Figure 2.

Figure 2. The Destination Computer folders.



The listbox in Figure 2 shows two components. The Always Installed component exists by default, and underneath it are two folders. The Application folder (which will be renamed Application Directory in a later beta) is the top-level directory for your install. You can specify a default location on the User Interface tab. Also under Always Installed is a Windows folder for common system files. I created a folder under Application called libsrc, and on the User Interface tab I checked Detect Clarion Location to default the application directory to the Clarion directory so that libsrc would match the user's libsrc directory.

I decided to make the template an optional part of the install (it isn't strictly required) so I created another component by right-clicking on Destination Computer. This component appears in Figure 2 as the Debugging Template folder. Under this I created a template directory.

To add the appropriate files to the libsrc and template directories I simply selected the files using the upper list boxes and clicked on Add Files. A later release will support drag and drop.

I then compiled the install program (which in this case took about two seconds on a P233 laptop) which resulted in a stunningly small 97k (108k in 32 bit) executable which included the classes and template, the install exe (which registered the template), and the uninstall program. This makes Setup Builder ideal for even very small installations. And given the product's ability to locate Clarion and register templates, it's a no-brainer for third party developers, or even for multi-programmer shops.

Adding password protection is as easy as typing in the password and clicking a checkbox. And you can automatically zip the resulting install program for upload to ftp sites (which may prohibit executables). For users currently working with Wise, Setup Builder will offer a Wise script import capability in the next beta.

As a beta, Setup Builder 3 still has a number of rough edges, and there will be at least several more beta cycles. Betas are coming out regularly, however, and there is an accompanying help file which is already better than some of the documentation which accompanies other Clarion third party products.

Setup Builder 3 will list for US\$99, but is free to all registered users of version 2, which sells for \$69. You do the math.

Copyright © 1999 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.