

**INVEST**  
in your own abilities**Clarion**  
magazine[Home](#) [COL Archives](#)**PDF for June, 2002**

All Clarion Magazine articles for June, 2002 in PDF format.

*Posted Monday, July 01, 2002*

**The Program's Finished - Now What? Part 2**

In Part 1, Braxton DeGarmo discussed the licensing of completed software to a publisher as a way of avoiding the hassles of marketing, sales, and support. Who wants all those headaches, right? Well, maybe you do.

*Posted Wednesday, July 03, 2002*

**A Single Browse For Multiple Lookups**

There have been a few threads in the newsgroups lately regarding techniques for handling a multitude of lookup tables in an application. Randy Rogers recently faced this situation, and came up with an efficient solution.

*Posted Friday, July 05, 2002*

**Creating A Dynamic Option Box**

Creating an option box is straightforward: select the option box, populate the radio buttons, compile and run. But users often want changes to option boxes that require a recompile. Steffen Rasmussen offers up a dynamically configurable option box, based on a list control.

*Posted Friday, July 05, 2002*

**Weekly PDF For July 1-6, 2002**

All ClarionMag articles for July 1-6, 2002 in PDF format.

*Posted Monday, July 08, 2002*

**Clarion/ASP Review Part 1**

Tom Hebenstreit begins an extensive, multi-part review of SoftVelocity's new Clarion/ASP product with some important background information, and a look at the install process.

*Posted Monday, July 15, 2002*

**Data Structures And Algorithms Part V - The Sort Goes On**

Alison Neal continues her discussion of sorting algorithms with the

**News**

[INN Bio & News for 29-Jul-2002](#)

[Server-Side PDF Generation](#)

[xTransparent Window v1.1](#)

[Dalby Source Printer Forum](#)

[ClarioNET Deployment Manager Available At ClarionShop](#)

[New Serial Communications Library](#)

[gFileFind 1.6 - Free Gift With Purchase](#)

[Clarion/ASP 1.1 Available](#)

[Nextage Security Templates](#)

[Screen Capture program](#)

[Dalby Source Printer Version 5.1](#)

[KSMsgBox - A Thread Safe Message Box](#)

[Firebird 1.5 Alpha](#)

[ImageEx 1.1 Released](#)

[Duke Application Shell Bundle/Pricing Changes](#)

[Icetips Takes Over CCS SQL Templates](#)

[SealSoft Update](#)

[LinderSoft News For July 2002 - Issue 1](#)

[New Lower SysAni Price](#)

[ABC Training With Bruce Johnson](#)

[Even More Free Software](#)

[INN Bio And News For 15-Jul-2002](#)

[Gitano 60 day Money Back Guarantee](#)

[EasyAutoEntry 1.00 Released](#)

Radix, Selection, Heap, and Quick sorts.

*Posted Wednesday, July 17, 2002*

## **Clarion Magazine Goes To Double Opt-In Email List**

This week marks a change in how Clarion Magazine sends out automated emails. Formerly, if you signed up as a member or subscriber, and simply indicated you wanted to receive email from Clarion Magazine, we sent it. This has now changed to a double opt-in process. Existing mailing list members should already have their update notices. If you didn't get yours, or you want to get the ClarionMag mailing lists, read on.

*Posted Monday, July 22, 2002*

## **Weekly PDF For July 14-20, 2002**

All ClarionMag articles for July 14-20, 2002 in PDF format.

*Posted Monday, July 22, 2002*

## **Clarion/ASP Review Part 2**

Tom Hebenstreit continues his extensive, multi-part review of SoftVelocity's new Clarion/ASP product. In this installment, Tommy creates his first Clarion/ASP web page.

*Posted Tuesday, July 23, 2002*

## **Custom Queue Sorting**

Clarion queues have two basic sorting methods built in - numeric, and alphabetical. But there are times when neither of those will suffice. Gordon Smith shows how to implement a custom queue sorting procedure.

*Posted Wednesday, July 24, 2002*

## **Welcome To The New ClarionMag Server**

You have reached Clarion Magazine at its new server location. If you've arrived here via the [www2.clarionmag.com](http://www2.clarionmag.com) or the 64.239.102.24 addresses, please do NOT adjust your bookmarks. Both of those aliases will be removed in a week's time. DNS changes should kick in within 24 hours anyway, and you will then be able to get here via the normal [www.clarionmag.com](http://www.clarionmag.com) address.

Dave Harms, Publisher

*Posted Wednesday, July 24, 2002*

## **Data Structures And Algorithms Part VI - All Sorted Out**

Alison Neal concludes her series on sorting algorithms with the Merge sort, and her personal favorite, the Shell sort.

*Posted Friday, July 26, 2002*

[0-Mezz Update](#)

[245 Products At ClarionShop.com](#)

[gFileFind Money Back Guarantee](#)

[Clarion/ASP Introductory Offer Extended To July 31](#)

[New ClarionShop.com Web Site](#)

[ImageEx Released](#)

[Bruce Johnson ABC Training In KC, Missouri](#)

[Flexible SQL Tables](#)

[The JaDu Dictionary Browser Returns](#)

[xAccess Manager Lite v1.3 Released](#)

[Updated DataModeller](#)

[SoftVelocity Holiday Hours](#)

[Free Inventory Software Updated](#)

[Taboga Software News](#)

[ABCFree Templates And Tools 2.41](#)

[Open Source Database Links](#)

[Remote Backup Code](#)

[ABC Bugs Listed](#)

[New dbZIP Templates](#)

[ASP Video 2](#)

[ASP Video 1](#)

[Icetips Professional Wizard New Build](#)

[MySQL Resources](#)

[Updated ImageMan Imaging Templates](#)

[Search the news archive](#)

## **Weekly PDF For July 21-27, 2002**

All ClarionMag articles for July 21-27, 2002 in PDF format.

*Posted Monday, July 29, 2002*

## **The Program's Finished. Now What? (Part 3)**

Braxton DeGarmo concludes his three part series on marketing with a look at the Internet as a marketing tool.

*Posted Wednesday, July 31, 2002*

## **Shrink-Wrapped Controls**

Changing the tab order of controls can be a real pain. Jeff Slarve shows how to make this tedious task easy with the window formatter's shrink wrap tool.

*Posted Wednesday, July 31, 2002*

## **Reports: OOP, ABC and Ignoring Templates (Part 1)**

While preparing his ETC presentation, "Report: Paying the Price of Flexibility," Steve Parker had an epiphany. He suddenly realized that he uses templates, well, *strangely*. Part 1 of 2.

*Posted Wednesday, July 31, 2002*

Looking for more? Check out the [site index](#), or [search the back issues](#).

This site now contains more than 700 articles and a total of over a million words of Clarion-related information.

Copyright © 1999-2002 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.

[Topics](#) > [Misc.](#) > [Marketing](#)

## **The Program's Finished - Now What? Part 2**

**by Braxton DeGarmo**

Published 2002-07-03

In my [last article](#), I discussed the licensing of completed software to a publisher as a way of avoiding the hassles of marketing, sales, and support. Who wants all those headaches, right? Well, maybe I do, and maybe you do, too. Am I saying this because I learned a lesson in my licensing venture? Actually, yes, but the lesson wasn't that I wanted to avoid licensing at all costs.

Not all programs you develop might be suitable for licensing to another publisher. Sometimes, your only realistic option is to become your own publisher. Perhaps the program has a very specific vertical market in which there is no dominant publisher, or you need a track record of sales to tempt a larger publisher's interest. Maybe you just want to retain full control of the product.

Whatever your reason, publishing can very rewarding, as long as you look at things positively. You know, like looking at tech support calls as a source of amusement rather than annoyance. Of course, if you've designed and implemented your program well and written excellent help files, maybe you won't be annoyed by too many calls.

So, now that you've decided to become a publisher, where do you go next?

### **Copyrights**

First, a disclaimer: the following is not legal advice. It is for informational purposes only.

Copyrights help protect you, to a degree, from illegal copying of your intellectual rights. That means your program, intellectual or not. Obviously, we all know how rampant piracy is in the software marketplace, so as a publisher, you may need all of the legal protection and help the

government can offer. By securing the copyright to your program, you have a tool to aid you in court should it become necessary to protect your product.

A copyright is “granted” to your program simply by its introduction into the marketplace. Under U.S. Copyright law, that usually means interstate or international sales. In court you would need to prove the date of the product’s release - its first, dated sales receipt will often suffice. Without such proof, the only valid documentation of copyright is that provided by formally registering your program with the U.S. Copyright Office, or the appropriate office in whichever country you reside and do business. (More information on that process in the U.S. is available at the Copyright Office’s [web site](#)) Even with dated invoices, it doesn’t hurt to have a federally issued copyright. The process is a simple one and shouldn’t require the services of a lawyer.

Okay, with your copyright in the process of being secured, what’s next?

## **Going To Market**

Marketing. There’s that word again. We’re programmers, not salesmen. Well, if you’re going to self-publish, you’re going to become a marketer and salesman as well. It goes with the territory. Of course, if you have the capital, you can hire someone with the necessary marketing expertise. But since a lot of Clarion developers are one- to five-person programming shops, I’ll continue on as if you don’t have a slickhaired, silvery tongued marketing or sales type in your employ.

Marketing can be simply divided into two tasks: identifying who might buy your program, and informing those potential buyers them about it. “Sales” is an altogether different undertaking. If your program is priced at impulse buying levels (under \$100 usually), there is usually very little involved in the actual sales process. On the other hand, if your product costs several thousand dollars, the sales process becomes complex, with product demos, phone calls and personal visits to convince the client that they need your program. You also need to close the deal and get that signed purchase order. But right now, I’m still talking about marketing.

The most critical aspect of marketing is identifying your market, those people who will buy your program. If you’ve developed software for a specific vertical market, this mission is often a lot easier. For instance, third party add-on developers for Clarion know exactly who their market is. Maybe your program is aimed at auto detailing shops, waste management companies, or a specific group of hobbyists. The narrower the market, the easier (usually) it becomes to identify potential customers. I say “usually” because if the market is in what is typically considered the general consumer marketplace, it may be more difficult to identify specific customers. In the business marketplace, many, if not most, vertical markets have trade groups and magazines aimed at their group. This makes it easy to identify an initial core group

of customers.

Once you have identified your target market, the next step is to analyze that market. What competition do you have and how well established are they in this market? How large is the market? What is the typical household income of people within this group? Or, if your target consists of businesses, what is the size of the typical company within this market? Are you dealing with publicly traded companies, individual proprietorships, partnerships, etc?

Get to know whom you'll be selling to. As an example, consider sport fishers. I developed an on-line database called CyberAnglers.com, and I needed to know my target market. In North America, there are 61 million sports fishermen, 40 million of whom fish for black bass. They average over 35 days of fishing a year, spending almost \$1200 a year on gear alone, and have an average household income of \$56,000 a year. How did I learn that? By looking into the magazines that cater to this group. Perhaps the best source of information on a specific market is the magazines that buyers/consumers in that market subscribe to. Magazines need to sell advertising to survive, and their media kits hold a wealth of information on their audience.

The same holds true for businesses. I have a cemetery management system that I developed for a local cemetery, while holding all rights to market the program to others. When I into the market research, I found that there are only 3500 active cemeteries in the U.S. Of that group, I was able to show that over 70% already had management software from one of eight or nine well-established competitors. With the price range of the competition, I decided it wasn't worth pursuing because the cost of marketing and sales would consume what profit I could expect.

So, what's the purpose of getting to know your target market? Simple. You can't target customers without knowing about them. And targeting is the next step, the means by which you will get your message across.

As stated before, if your market is a specific vertical market, you have a limited number of ways to approach them with your advertising: trade magazines, trade organization mailing lists for direct mail, trade shows, broadcast fax, and the Internet. For general consumers, the Internet alone might be your only avenue of reasonable expense.

I have no degree in marketing except from the School of Hard Knocks and almost twenty years experience in marketing and selling my own software to a variety of markets. As I see it, there are four major forms of marketing: print advertising, broadcast media (radio and television), person-to person (trade shows and door-to-door) and direct mail. I personally see the Internet not as a new form, but as another avenue for three of these forms, with banner ads falling under print advertising, websites offering both print and broadcast features, and mass email (preferably opt-in mailings) as direct mail. (Broadcast media is only appropriate for very



localized markets and should be considered only in a local or regional marketing campaign to the general public.)

Which is best? That depends on your market and your product. If you are selling a several hundred or thousand dollar system to a specialized market, perhaps you need the prestige of full color print ads in one or more trade journals, combined with full color brochures mailed to a list purchased from that market's trade organization. But if your market is small business or hobbyists, print advertising will never pay for itself. It will rarely generate the volume of leads needed to translate into enough sales to receive a satisfactory return on investment (ROI). For that market, direct mail might be a better avenue, if you can purchase mailing lists from one or more magazines dedicated to your group's interests.

## Calculating your ROI

Whichever form(s) of marketing you choose, you need to track your ROI for each method. Determining your ROI is a five-step process.

**Step one** is to calculate the Conversion Rate (CR). This is the number of purchasers divided by the number of "ad impressions." With direct mail, a 1-2 percent return is considered good. So if you mail to ten thousand people, you might get 150 responses. Of those, say 50 actually buy the program. That's a CR of  $50/10000$ , or .005.

**Step two** is the "impressions per sale" ratio:  $1/CR$ . In this example, the CR is  $1/(.005)$  or 200. So you need 200 people to see your ad to generate one sale.

The Marketing Investment (MI) is **step three** and is calculated as (the per capita price of the ad) X (impressions per sale). So, if your direct mail piece costs \$.75 each, the MI is  $(0.75 \times 200)$  is \$150 per sale. You can see already that if your product doesn't cost at least \$150, you're in trouble. For print and broadcast advertising, the ad people will give you their best estimate of ad impressions for your print ad or radio spot. By ad impressions, I mean the number of people who will be impressed by your ad. Well, a guess as to how many might actually see or hear it anyway. Divide this by the total cost of the ad to get a per capita number.

The **fourth step** is to determine your profit: Price – Cost – MI. Let's assign a cost of \$299.95 to the product. (Isn't that about the average cost of a Clarion add-on?) And this product has a cost of \$10 to produce. The profit becomes:  $\$299.95 - 10 - 150$ , or \$139.95.

The **final step** is to calculate the ROI, which is the Profit/MI ratio. In this example it's \$.93. In other words, for every dollar spent on this direct mail campaign, you received \$.93 back. So, despite making a profit on each sale, you still lost money on that advertising campaign. By reducing the cost of the mailing to just \$.70 each, however, the ROI becomes \$1.76, which is a profit on each dollar spent.

Of course, the Internet has altered forever the task of marketing. With it, you can reach millions of potential customers for a fraction of the expense of traditional marketing. However, the keyword here is potential, because you are also one of millions now using the Internet to market a product line. The challenge of the Internet is one of being found by those potential customers. In my [next article](#), I'll explore the Internet as a marketing medium.

---

*[Braxton DeGarmo, MD](#) has been programming with Clarion since 1989. As a practicing physician, he needed the most efficient development tool possible to continue his medical programming, and Clarion was the perfect prescription. In the past two years, he has broadened his programming to several vertical markets, including sportfishing and fiction writing.*

## Reader Comments

[Add a comment](#)

**Just a quick comment to let people know that the two...**

**I am curious what the author and readers would submit to...**

Copyright © 1999-2002 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.



**Reborn Free****CLARION**  
*online*[Home](#) [COL Archives](#)[Topics](#) > [Browses](#) > [Browses, Using](#)

## **A Single Browse For Multiple Lookups**

**by Randy Rogers**

Published 2002-07-05

There have been a few threads in the newsgroups lately regarding techniques for handling a multitude of lookup tables in an application. I am sure most Clarion programmers have been faced with the same dilemma. After normalizing your database you find you have many tables that are basically the same: a code and a description. It seems like such a waste to maintain individual files for these typically small tables.

In the past I have take the approach of combining all of these lookups into one table with an additional field to segment them. This minimizes the number of tables, but generally requires that the segmentation field value be hard coded into the application. This can be a hindrance to users who might want to use a report writer or other query tool to view the database, since they would have to know the segmentation codes in order to properly join the lookups to the primary table.

I recently started a project that had to adhere to a predefined database structure. There are certainly parts of the database that I would have defined differently (like not joining tables on data elements that are modifiable by the end user), but it is a fairly normalized data specification. This particular database contains over 250 tables of which about 175 are lookup tables.

The lookup tables all have the same basic layout, a four character code and a 64 character description field. I am not particularly adept at getting the wizards to generate these lookup browses for me. I can get the basic list box and update buttons, but still have to go into each browse and fine tune it to be exactly the way I want it to be. After tuning ten browses at about 30 minutes each, I calculated it was going to take me almost two weeks to complete the lookups. I was already bored with the repetitious work and thought there had to be a better way; necessity is, after all, the mother of invention.

Clarion's template language, in my opinion, has always been one of the best features of the tool. I haven't mastered the wizard paradigm but I am pretty good at creating templates. My plan was simple: compare two browses to discover the differences, then create a common browse procedure that would receive the differences as parameters and use edit-in-place to update the fields. Once the common browse was completed, I intended to create a template that would generate the code to call the common browse for each of the lookup tables. The result had to work just as if I had created the 175 lookup browses in the normal fashion.

This approach took me about a day to implement, and the results were so good that I thought it should be shared with the Clarion community. At the end of the day, I had completed all 175 lookups. I only had one browse to maintain, I could add new lookups without additional coding, and the size of the DLL was reduced from 1200 kb to 650kb. The remainder of this article details the technique.

## One browse for all lookups

I started by comparing two of the browses I had already fine tuned. What I discovered was that each browse required a caption, a view, a relation manager, a key, and pointers to the code and description fields. Armed with this list of differences I took one of the generated browses and used it as the starting point for a hand coded source procedure. When creating the procedure prototype, I used parameter names that matched the labels that had been created by the generator; this was to minimize the code changes I would need to make. Here's what the final prototype looks like:

```
Browse:LookupTable PROCEDURE (*cstring szCaption, |
    VIEW BRW1::View:Browse, RelationManager    RM, |
    KEY K, *? szField1, *? szField2)
```

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

```
Queue:Browse    queue
Id              cstring(5)    !code field is 4 bytes
Description     cstring(65)    !description is 64 bytes
Mark           byte      !Entry's marked status
ViewPosition   string(1024)  !Entry's view position
end
```

I only needed to make a few more changes in the `WindowManager.Init` code after the `open(window)` call.

```
OPEN(Window)
SELF.Opened=True
```

```

window{prop:text} = szCaption
BRW1.Q &= Queue:Browse
BRW1.FileLoaded = 1
BRW1.AddSortOrder(,K)
BRW1.AddLocator(BRW1::Sort0:Locator)
BRW1::Sort0:Locator.Init(,szField1,1,BRW1)
BRW1.AddField(szField1,BRW1.Q.Id)
BRW1.AddField(szField2,BRW1.Q.Description)
SELF.SetAlerts()

```

The final modification was to the `BRW1:Init` procedure. Since I had already used the label `RM` for the relation manager parameter to the browse, it was necessary to change the parameter name used in the `BRW1:Init` code from `RM` to `RelMgr`.

```

BRW1.Init PROCEDURE(SIGNED ListBox,*STRING Posit,VIEW V,|
    QUEUE Q,RelationManager RelMgr,WindowManager WM)
CODE
PARENT.Init(ListBox,Posit,V,Q,RelMgr,WM)

```

Next, I needed to test this new `BrowseLookup` procedure, so I hand coded a small test procedure. The code looked like this:

```

Browse:AGP_TYPES    PROCEDURE
szCaption    cstring('Browse Artificial Gravel Pack Types')
View:Browse view(AGP_TYPES)
    project(AGPT:WL_AGP_TYPE)
    project(AGPT:WL_AGP_TYPE_DESC)
end
CODE
Browse:LookupTable(szCaption, View:Browse, |
    Relate:AGP_TYPES, AGPT:PK_AGP_TYPES, |
    AGPT:WL_AGP_TYPE, AGPT:WL_AGP_TYPE_DESC)
RETURN

```

I quickly compiled the application and gave it a test: it worked like a champ! And the little test procedure only contained two data declarations and a call to the `Browse:LookupTable` procedure. Now I was ready to create the template.

## The extension template

The first hurdle was to determine which files in the dictionary were lookup tables. I could have done this by defining a user option for each of the lookup files. But, being a lazy programmer, I decided this would take too much effort. I knew all the non-lookup tables in the dictionary had more than two fields, so I created the template to generate code for all files with only two. The generated procedures also needed to be added to the map and export list. My intent was to add this template to my hand-coded `Browse:LookupTable` procedure, so the actual procedures are generated into the `Processed Code` embed.

```

#!-----
#TEMPLATE(CommonLookup, ◀
    'Common Lookup Generator'),FAMILY('ABC')
#!-----
#!
#!
#SYSTEM
#DECLARE (%Field1)
#DECLARE (%Field2)
#DECLARE (%LookupFile),MULTI,UNIQUE
#DECLARE (%Module)
#!-----
#EXTENSION(GenerateLookups, ◀
    'Generate Lookup Browse procedures')
#!-----
#! Create list of lookup files (only two fields)
#ATSTART
    #FOR(%File)
        #SELECT(%Field,3)
        #IF(NOT %Field)
            #ADD(%LookupFile,%File)
        #ENDIF
    #ENDFOR
#ENDAT
#!
#! Add the procedures to the map
#AT(%GlobalMap)
#SET(%Module,UPPER(%Module))
MODULE('%Module')
    #INDENT(-5)
    #FOR(%LookupFile)
        #SET(%ValueConstruct,'Browse:' & %LookupFile)
%[32]ValueConstruct PROCEDURE
    #ENDFOR
    #INDENT(5)
END
#ENDAT
#!
#! Add the procedures to the export list
#AT(%DLLExpertList)
    #FOR(%LookupFile)
        #SET(%ValueConstruct,'BROWSE:' & %LookupFile & '@F')
%[55]ValueConstruct @?
    #ENDFOR
#ENDAT
#!
#! Generate the Browse:LookupFile procedures
#AT(%ProcessedCode),PRIORITY(7500)
#FOR(%LookupFile)
#FIX(%File,%LookupFile)
!-----
Browse:%File    PROCEDURE
!-----
szCaption    cstring('Browse %FileDescription')
View:Browse view(%File)
    #FOR(%Field)

```

```
        project(%Field)
#CASE(INSTANCE(%Field))
#OF(1)
    #SET(%Field1,%Field)
#OF(2)
    #SET(%Field2,%Field)
#ENDCASE
#ENDFOR
        end

CODE
Browse:LookupTable(szCaption, View:Browse, |
    Relate:%File, %FilePrimaryKey, |
    %Field1, %Field2)
RETURN
#ENDFOR
#ENDAT
```

I registered my new template and attached it to the `Browse:LookupTable` procedure, then compiled the DLL. Finally, I added menu options to my application frame to call each of the generated browses and compiled the application. I demonstrated the results to the rest of the team, then told the boss I was taking two weeks off since I had my work completed! Not really, since I am the boss, but we did joke about the time off.

Since that time I have added about five more lookup tables to the dictionary. A simple recompile of the DLL and addition of application menu items is all that is required to add additional lookup tables to the project. If we want to modify the look and feel of the lookup browses, we only need to change the one browse procedure and it is done. The entire application is considerably smaller because of this approach, and we have relieved ourselves of the repetitious and boring task of fine tuning wizard-generated browses.

[Download the source](#)

---

*[Randy Rogers](#) is a data processing professional with over 35 years of experience in a wide variety of industries including accounting, municipal government, insurance, printing, and pharmacoeconomics. He is the president of Keystone Computer Resources and creator of NetTools, Queue Edit-in-Place and Screen Capture Tools for Clarion application developers. Randy has a degree in Mathematics from Florida State University and has taught programming at the community college level.*

## Reader Comments

[Add a comment](#)

**I downloaded the source and it won't compile (lute...)**

**I keep getting "ADDCELL@F14RTFWRITERCLASSI Is unresolved...**

**This is not a problem with the example app - it's because...**

Copyright © 1999-2002 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.

**INVEST**  
in your own abilities**Clarion**  
magazine[Home](#) [COL Archives](#)[Topics](#) > [Design & Development](#) > [User Interface](#)

## **Creating A Dynamic Option Box**

**by Steffen Rasmussen**

Published 2002-07-05

Creating an option box is straightforward: select the option box, populate the radio buttons, compile and run. That's until your users start to use your program. They all have these great ideas, like changing the text for one or more of the radio buttons, or adding new/deleting existing radio buttons. They may even want different radio buttons based on a certain condition. And the best part of it all is that during development they don't have all these ideas.

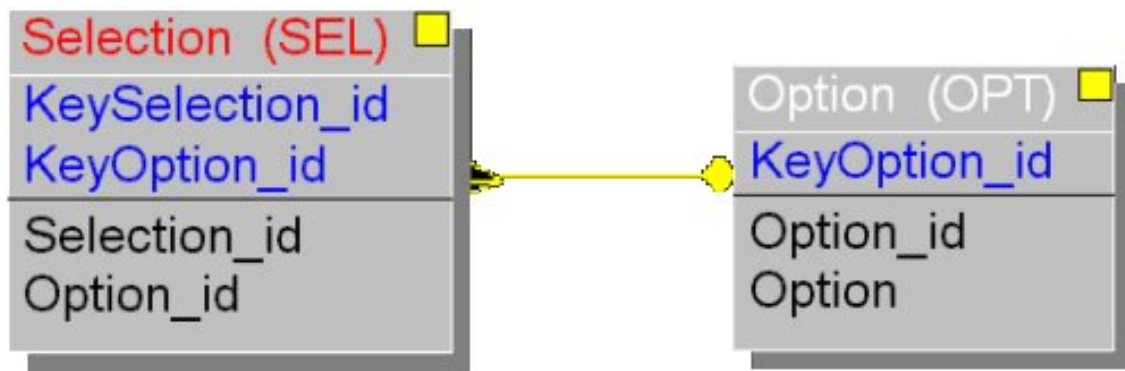
Clearly the users first really know what they want and how to improve it when they have had hands-on experience with your product. So you have to make "minor" adjustments. The only difference is that the amount of work you have to do in order to make these changes has increased dramatically, because now you also have to redistribute your application once again. Of course you could wait until the next update, but that really won't satisfy your users.

An alternative solution is to design your option boxes in such a way that you can change them at runtime. In this article I'll show you how to do that by using a list box to mimic an option control.

### **The dictionary**

As with most application creation in Clarion, the work starts in the dictionary. You're going to create a list box that looks like an option. That means that instead of defining the radio buttons within a field in a table, you have to split the contents of the radio buttons into a separate table, which is related to the table that contains the value of the selection (as shown in Figure 1). In this way you can put as many records into the list box as you like. You will, however, want to keep the number to a minimum so the user doesn't have to scroll the list box. You are, after all, creating an option box.





**Figure 1. Entity relations diagram**

## Designing the Option Box

In the update procedure where you want to use the option box, start by creating a local variable called LOC:RadioButton. Use BYTE for the data type and @p p for the Screen Picture. This variable is used to contain the icons representing the radio buttons.

```

IF LOC:InitListbox = 0 !IF first selection
    %Control{PROP:Selected}= ForeignKeyField
    LOC:InitListbox = 1
ELSE
    %ForeignKeyField = %PrimaryKeyField
END
  
```

**Edit Column Properties - LOC:RadioButton**

General | Attributes | Comments | Options | Help | Validity Checks | Window | Report

Column Name: LOC:RadioButton

Derived From: [ ] [...]

Description: [ ]

Data Type: BYTE [v] Prefix: [ ]  Reference

Base Type: [ ] [...]

Characters: [ 1 ] Places: [ 0 ]

Dimensions: [ 0 ] [ 0 ] [ 0 ] [ 0 ]

Row Picture: [ ]

Screen Picture: @p p [...]

Prompt Text: LOC : Radio Button:

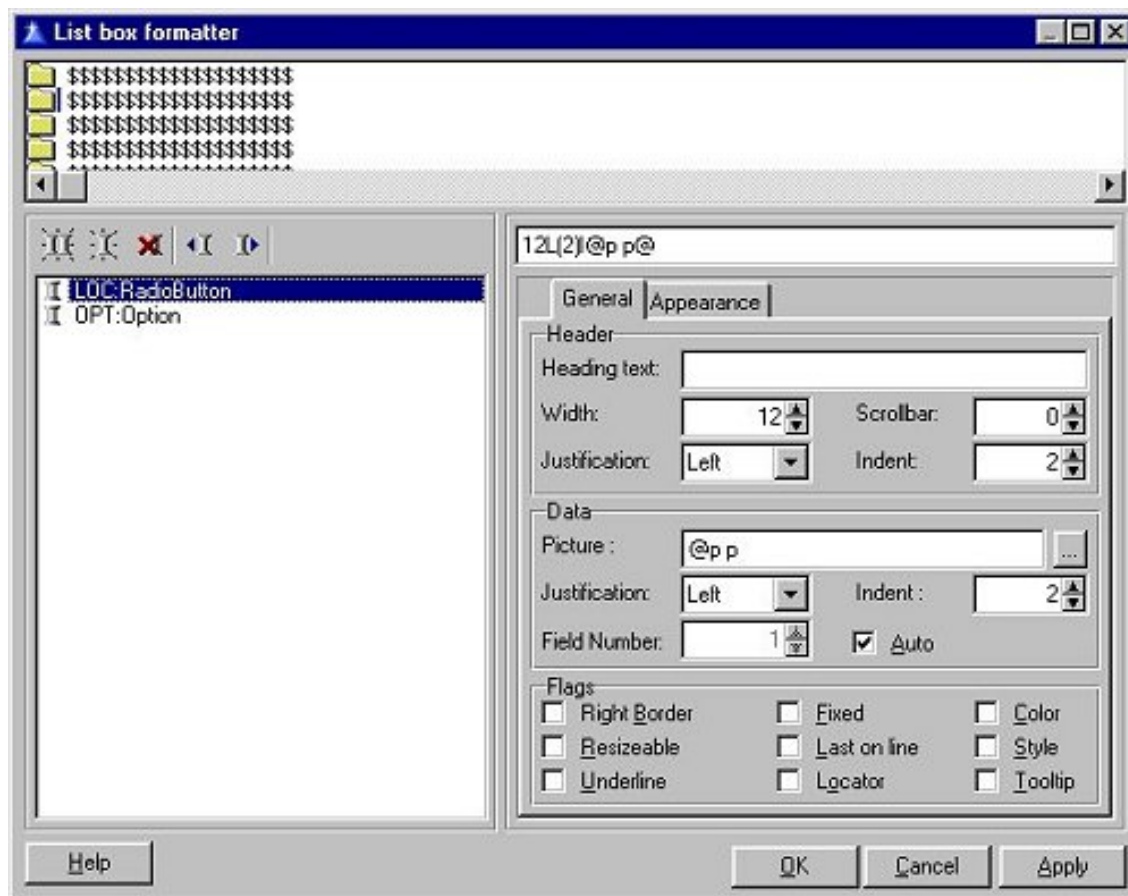
Column Heading: LOC : Radio Button

Freeze

OK Cancel Help

**Figure 2. Edit column Properties**

In the Window Formatter create a list box based on the table containing the option box selections. The list box should only contain two columns. The first column contains the radio buttons for the option box selection. Here you use the local variable you just created. There shouldn't be any headers, so on the General tab delete the Heading text so it is blank. In the Flags group make sure that all of the check boxes are unchecked.

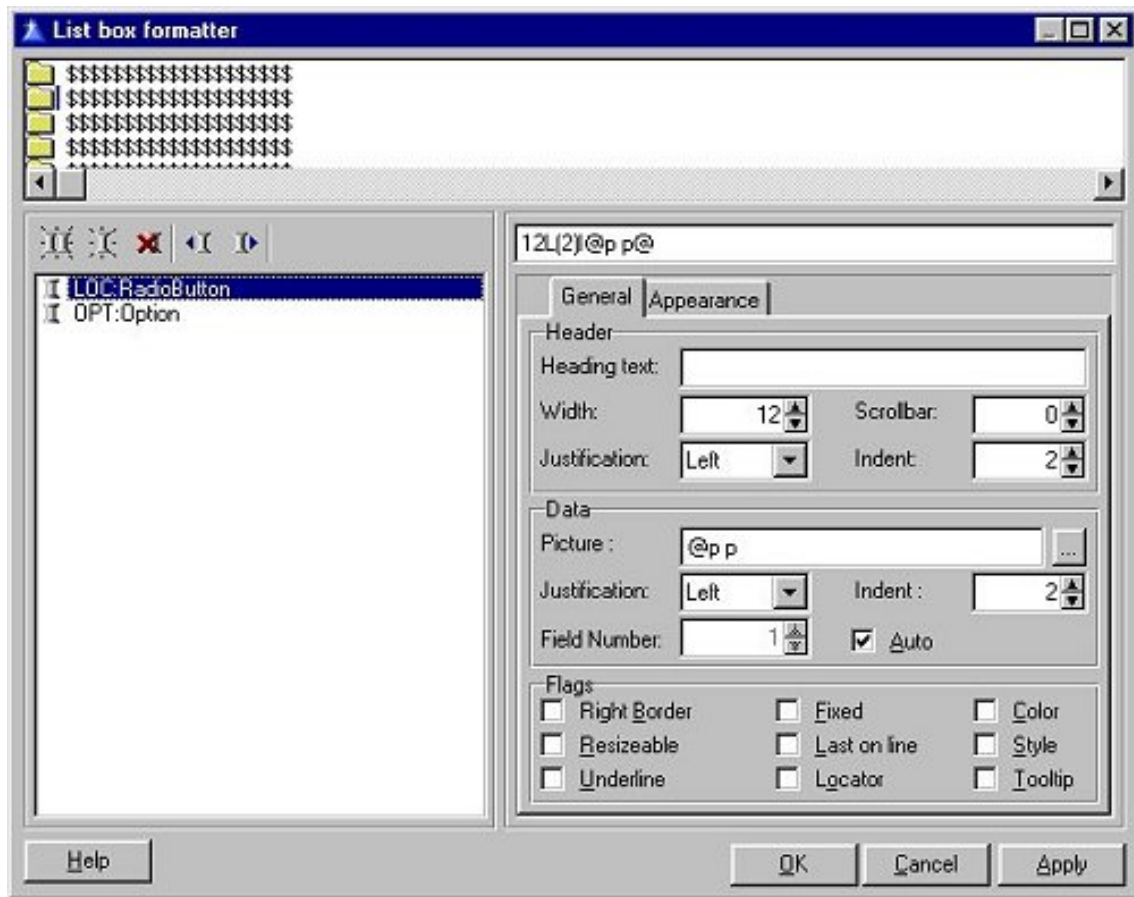


**Figure 3. List box formatter**

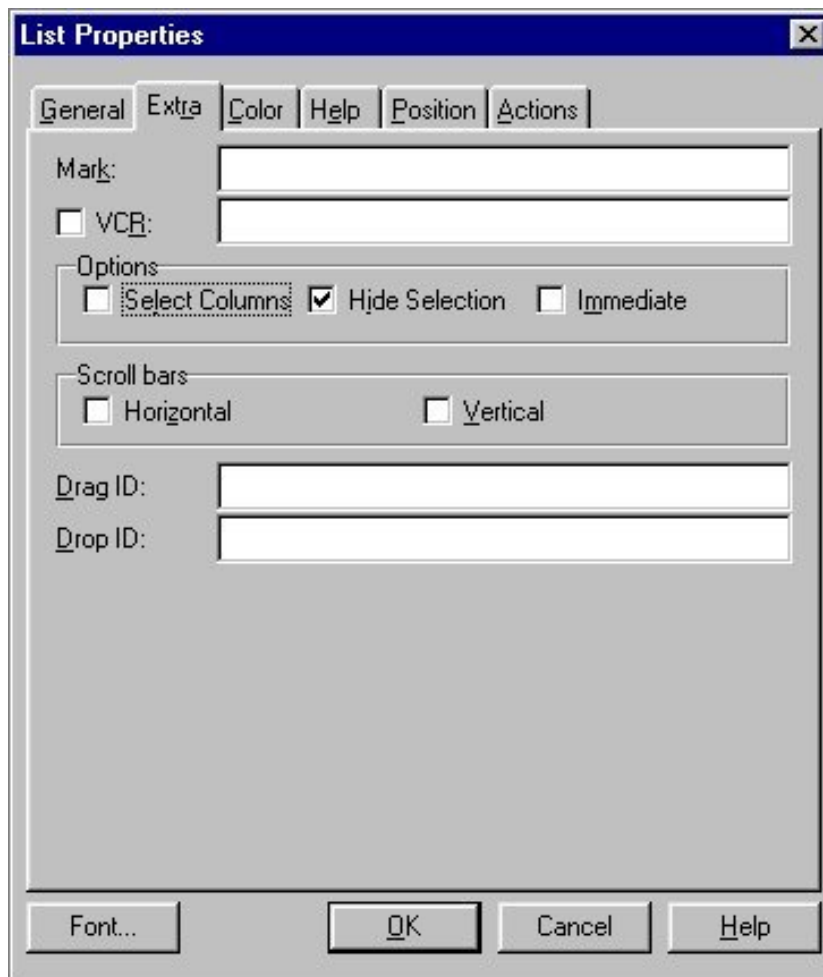
Now select the Appearance tab, and in the Icon group select Normal. This creates a column which only contains an icon.

The second column contains the text of the radio control. Like the first column you should delete the heading text as well as clear all the flag settings. In the Data group set the justification to left. Set the Indent field value according to how far from the Radio button icon the text should be. I prefer to use an indentation of 2. On the Appearance tab make sure that the Icon group is set to None. Press OK to accept the changes in the list box editor.

Next right click the List Box and select Properties. On the General tab, in the Mode group, select Transparent. This makes the list box background disappear.

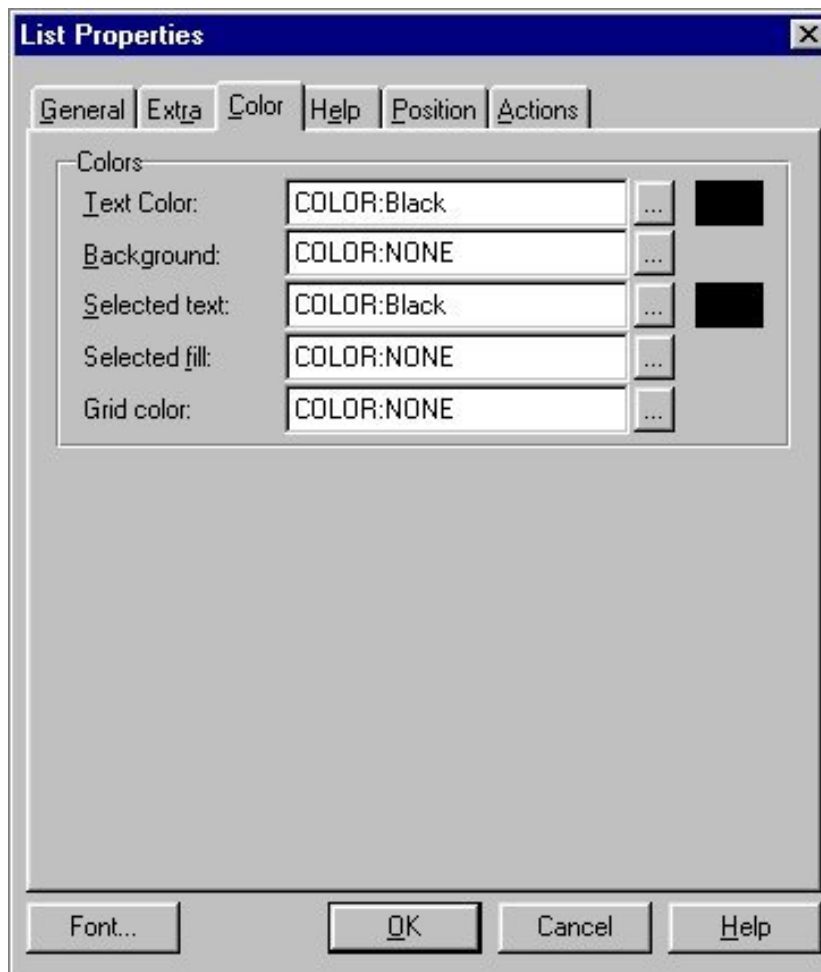


**Figure 4. List box General properties**



**Figure 5. List box Extra properties**

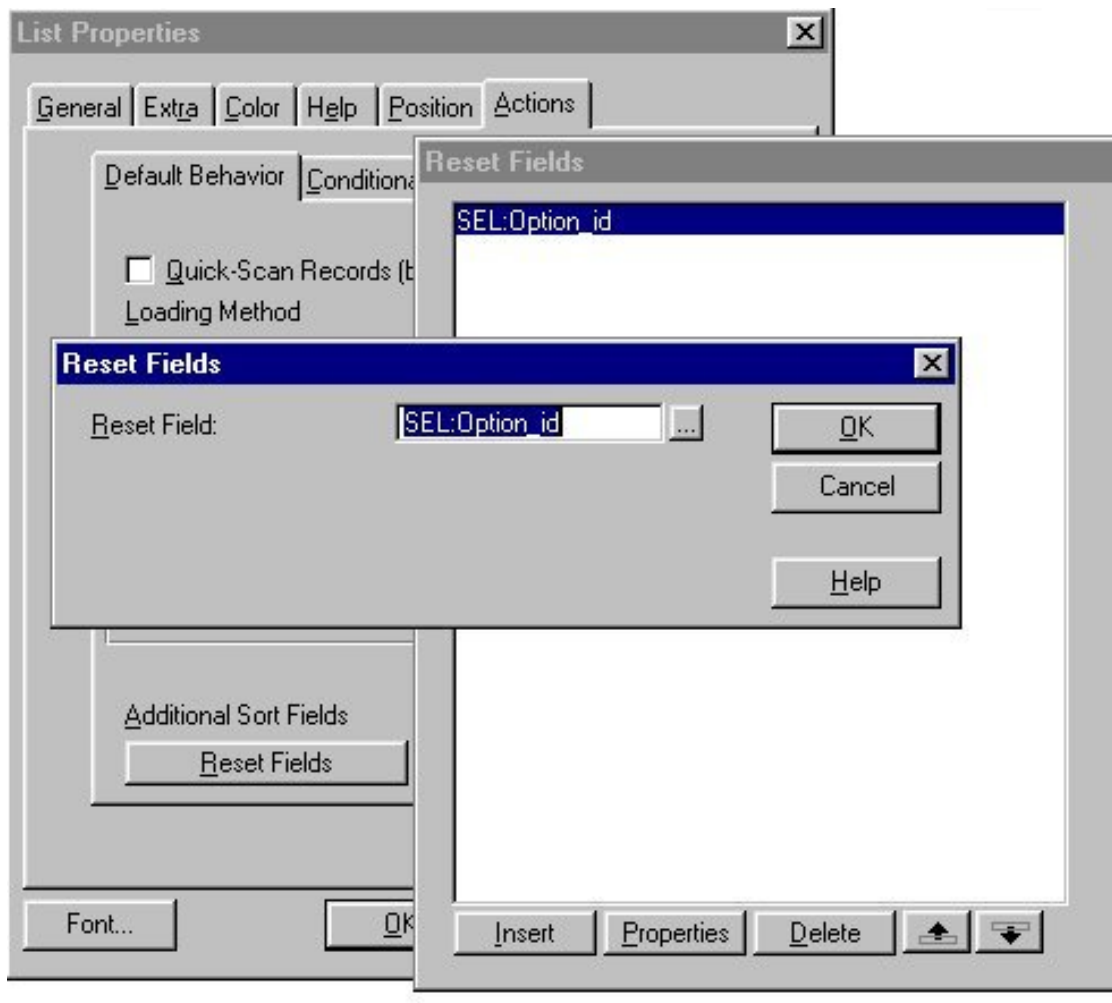
Select the Color tab. Here you have to select a color for the selected text so the text color does not change when the user selects an option. In this example the text is black.



**Figure 6. List box Properties Color**

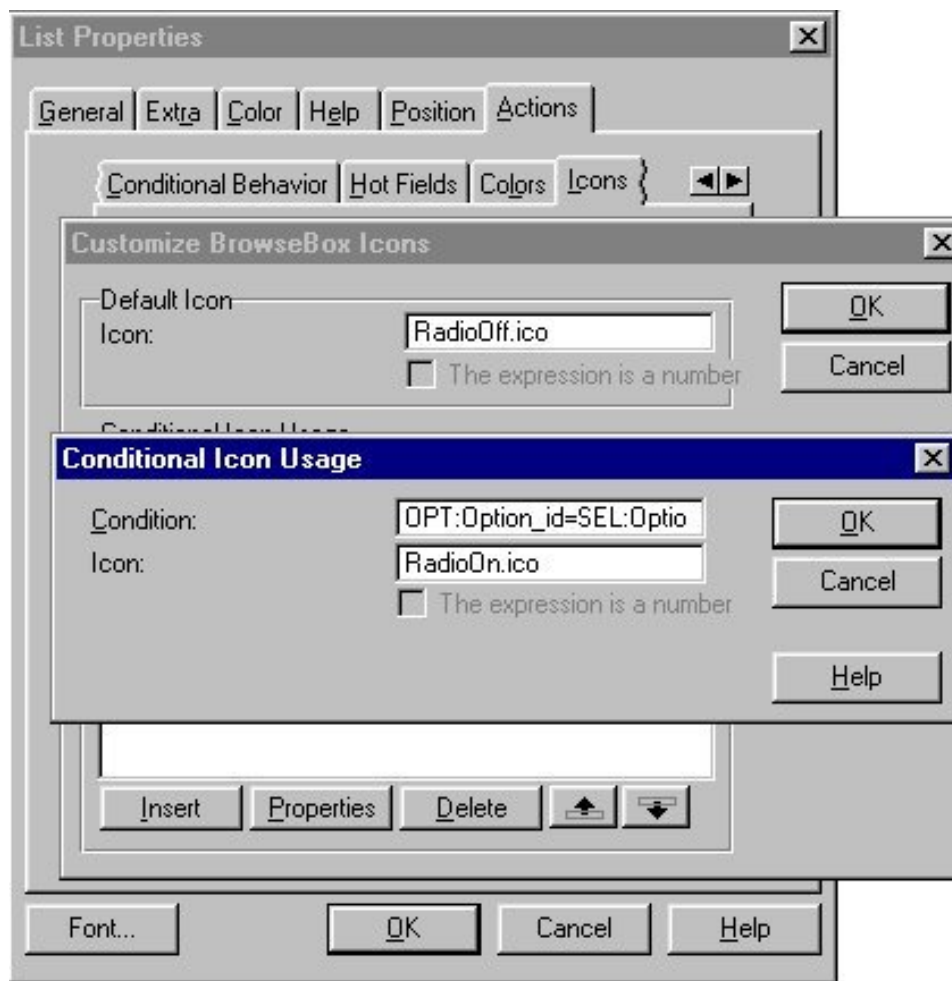
On the Actions tab you have a number of options to select from, depending on how complex you want the relations and conditional behavior to be. One thing that's not optional is the Loading method, which has to be set to File.

Among the options that you can set is the possibility to make the options change at runtime. To do this you will have to set the Reset Field. The reset field is used to monitor for any changes in the foreign key field relating to the table with the option box values. When a change occurs in the reset field, the ABC code will refresh the list box so it reflects the new selection by changing the icon in the List box.



**Figure 7. Reset fields**

To be able to change the Icons based on the selection, you have to set a condition for the Icon usage. Select the Icon tab and Insert. In the Default Icon group write: `RadioOff.ico`. This is the icon for a radio button which is not selected (it is included with the example application). In the Conditional Icon Usage group select Insert. Here you have to set a condition in order to make the Icon change. The condition is really simple - all you have to do is compare the field in the file you're updating, which stores the list box/option box value, with the primary key field in the option box table. If there is a match, then the user has selected a field in the list box, in which case the icon `RadioOn.ico` has to be set.



**Figure 8. Conditional icon usage**

## Making it all work

The last thing you will have to do is to program the list box selection. So for each new selection the selected primary key field value is assigned to the corresponding foreign key field. That's easy, right? Just embed code that accomplishes what the following pseudocode does at the `TakeNewSelection` embed point for the listbox:

```
ForeignKeyField = PrimaryKeyField
```

Unfortunately this doesn't quite work, because every time you open the procedure containing the option box the first record in the list box is selected and assigned to the foreign key field. If it is a new record you are creating this behavior could be accepted, although it doesn't follow the option box paradigm, which is to not select any option unless the selection has been explicitly primed. If on the other hand you're updating a record, this behavior is not acceptable because you want the option box to show the previous user selection, until a new selection in the option box has been made.

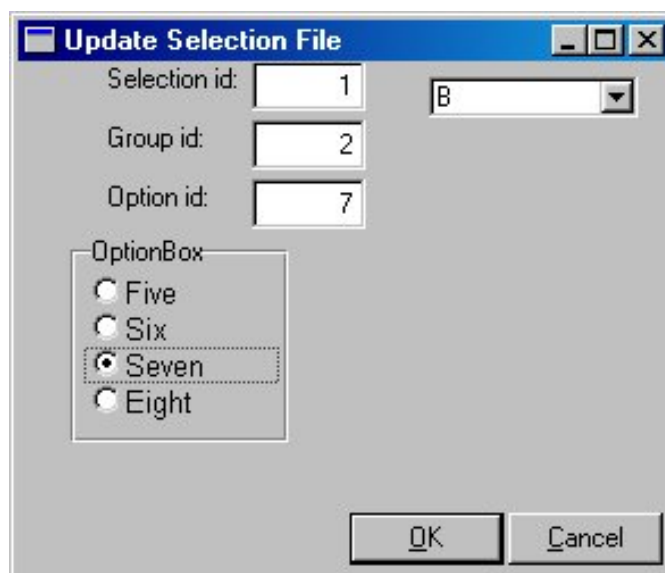
Since the list box shows a marked radio button when the `ForeignKeyField = PrimaryKeyField` you have to force a list box selection so `PrimaryKeyField = ForeignKeyField` when the first



selection is made in the list box by the user. To do this you could use a local variable `LOC:InitListbox`, which initially is zero. When the first selection is made the variable is changed to one:

```
IF LOC:InitListbox = 0 !IF first selection
    %Control{PROP:Selected}= ForeignKeyField
    LOC:InitListbox = 1
ELSE
    %ForeignKeyField = %PrimaryKeyField
END
```

That's all there is to making a dynamic option box out of a list box. Figure 9 shows the such an option box in the example application.



**Figure 9. The dynamic option box**

To make things even easier, I've included an extension template in the example application. All you have to do to use it is to select the Extension button in the procedure properties window. Now select the list box which is used as an option box and press the Insert button. Select the `BrowseRadioButton` extension in the Class Tools subfolder. In this extension template you have two entry fields which have to be populated, the same way as you would with the `PrimaryKeyField` and the `ForignKeyField` in the code above. So instead of using the above code you can implement the template and populate the other list box/option box properties as described.

[Download the source](#)

---

*[Steffen S. Rasmussen](#) has graduated in Computer Science from Copenhagen Business College. Since then he has worked as a programmer, system technician and network administrator, and is currently IT manager. Clarion is a quite a new language to*

*Steffen since his only been working with it since January 2000. But what better way to learn it than by trying to teach others!*

*Steffen has also set up a [web site](#) to collect as many examples of different user interfaces as possible to inspire Clarion developers.*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2002 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.

For marketing your Applications  
and Developer Accessories or to  
purchase other 3rd Party Tools . . .

Developer **PLUS**<sup>tm</sup>

[Home](#) [COL Archives](#)

[Topics](#) > [Reviews](#) > [Reviews](#)

## **Clarion/ASP Review Part 1**

**by Tom Hebenstreit**

Published 2002-07-15

Strike up the band, folks, for Clarion/ASP has arrived!

Clarion/ASP is a milestone product for the SoftVelocity and the Clarion community in a couple of important ways: It is the first product wholly developed by SoftVelocity, and it is the first Clarion product that has absolutely nothing to do with Clarion the language.

What exactly is Clarion/ASP? It is a set of templates that use the Clarion data dictionary and IDE as a framework to generate pure web applications. No Application Broker is required, no Clarion code is used and no Windows programs are created – only HTML and ASP pages (which are just text files containing ASP source).

At first I had planned on doing this review as a single article, but the more I delved into Clarion/ASP and all of the associated technologies, the more I realized what a deep product it is (and how long it might take to write the review!). And Clarion/ASP is also the first product to be accompanied by the new SoftVelocity online training classes, so I'll be covering that as well. The upshot of all of this is that rather than waiting to do this review until I have completed the online courses and fully learned the product, I'm going to spread the review over a number of weeks, and let you follow my adventures in this brave new world of ASP, ADO, SQL and IIS from the beginning.

I will break the review into these broad areas:

- Requirements, terminology, installing and getting started
- Online training, building and testing applications
- Integration, customizing generated code and summary

SoftVelocity has stated that they are planning on releasing a maintenance update of Clarion/ASP within the next couple of weeks to address some of the initial teething problems and to add new functionality requested by the early adopters, so I'll be able to cover that as well during the course of this review.

Let the fun begin!

## Essential terminology

Since Clarion/ASP ventures outside of the comfortable confines of the Clarion language and into a sea of acronyms, I'll first define a few of the more essential terms. Note that these are simplistic definitions, and I will try to relate them to their logical counterparts in Clarion to get the basic concept across. There is much more detailed information available: if you piled all of the books written on IIS, ASP, ADO and SQL one on top of the other, the resulting stack would probably be taller than a ten story building.

**IIS** – Internet Information Services. This is the free web server that Microsoft includes in Windows 2000 and XP (or as part of the Option pack for NT). IIS provides the environment that allows you to actually run your Clarion/ASP applications.

**ASP** – This stands for Active Server Pages, and it is the basis for the existence of Clarion/ASP. It is a method, developed by Microsoft, for creating web pages on the fly using a combination of standard HTML tags and special scripting (i.e., programming) languages. In a broad sense, you can think of each ASP page (file) as a mini-program unto itself that is interpreted and run on the spot by IIS whenever someone requests that ASP page from the web server. And, as the IIS web server executes the ASP code, the scripts dynamically create the output that is sent back to the browser to be displayed. For example, you could use an ASP page to fetch and format a set of records from a database.

**ADO** – ActiveX Data Objects is Microsoft's preferred method for accessing databases in Windows (indeed, any data source whether SQL, flat files or even an Excel spreadsheet). ADO is built upon COM, and can either access your data source directly (if the vendor has built an OLEDB provider) or via the standard ODBC layer. In Clarion/ASP you can think of ADO as the rough equivalent of a combination of an ABC FileManager, ViewManager and a file driver. ADO objects are created and used by the ASP scripting language (i.e. VBScript) to handle the details of opening, closing, fetching, updating and otherwise navigating through your database.

**SQL** – Structured Query Language. As most people have already at least heard of SQL, I won't go into it in detail here other than to point out that SQL is normally the syntax that you use to talk to your data source via ADO. If the back end doesn't understand SQL, it is the job of the underlying provider or ODBC driver to translate the SQL into instructions compatible

with the data source.

So, in a nutshell, IIS implements ASP which uses SQL via ADO to access your data. Got it? Good.

## Major features

I usually list off a bunch of features in this space, but to save both time and space I'm just going to point you to the [Clarion/ASP portion](#) of SoftVelocity's web site. As a demonstration about how serious (and proud) SoftVelocity is of Clarion/ASP, there are already extensive downloadable white papers, video demos, example Clarion/ASP applications and more. Also, since there will be changes in the product due to the pending update, I don't want to waste time listing information that could be wrong in a few weeks!

## Getting ready

Clarion/ASP is a different kind of beast when it comes to getting an application ready for use, as there are four distinct environments that must be properly set up in order for it to work:

**Windows** – A word of advice: Save yourself a lot of trouble and do your Clarion/ASP development under Windows 2000 or XP (Professional, not Home). You can use Windows NT 4, but you may have to update a few components (you'll also have to put up with the wretched NT user interface.) Using Windows 2000 or better, most of the required bits such as ADO and IIS come pre-installed and ready to go (too ready, in some cases – see the IIS topic below.)

**Clarion** – The templates require Clarion 5.5h (otherwise known as 5.508). This latest version installs new features that are used by the Clarion/ASP templates, such as the Connection Builder (a utility to help you build the string of instructions used to connect to your database). In a pinch, the templates will work with 5.5g (5.507), but you will lose out on Connection Builder. Earlier versions will not work with Clarion/ASP at all.

What to do if, like me, you don't want to (or simply can't) upgrade your production systems to 5.508? Other than using multiple machines for the different versions, the simplest trick is to just rename your production \C55 folder to, say, \C55NonASP and reinstall the product. Patch the new version to 5.508, and then use that exclusively for Clarion/ASP work. When you need to use the older version for Windows programming, just rename the two folders again. You could set up a batch file to switch back and forth. I have multiple version of Clarion 5.5 installed on my machines in this manner and it works quite well. A side benefit is that your Clarion/ASP/Clarion 5.5h combination will be very quick to load, since no third party or other templates need to be installed for ASP generation.

**ADO and a Database** – First off, since Clarion/ASP does not use the Clarion language or

create Windows programs, it logically follows that it does not (cannot) use the normal Clarion database drivers. As I mentioned above, ASP uses ADO to connect to your data source and to handle all of your database functionality, and that means you must have ADO (and ODBC) installed/available in three places: Your Clarion development environment, the web server and the database server. These can be individual dedicated machines on a network, or all three components can reside on a single machine. If, for some reason, you do not have ADO and ODBC installed, they are part of MDAC (the Microsoft Data Access Components), which can be [downloaded](#) from Microsoft.

The second part is that you must decide what database/data source you are going to be using. Although you can use just about any data source via ADO, SoftVelocity recommends that you use a SQL database of some sort, and I concur for three reasons. First, a web application should be as robust and scalable as possible. Starting off with a SQL backend lets you scale upwards, hopefully without having to change anything in your application. Secondly, SoftVelocity has oriented a great deal of their training examples to using MS SQL Server (with some nods towards Access MDB files if you don't have SQL Server). Thirdly, I know the temptation is there to start off using a familiar format like TPS files, but the plain fact is that the ODBC driver for TopSpeed files is not the most full featured and stable beastie in the world (there *are* people using it with Clarion/ASP, though). It's your choice, but I strongly recommend SQL.

## **The web server**

You must have administrative access to an IIS web server (or someone who does) so that you can set up the directories where your applications and images will reside. You also will need to transfer your generated Clarion/ASP pages to the server in order to test them.

## **Putting it all together**

Here are my personal recommendations for the most painless way of getting started with Clarion/ASP:

- A clean Clarion 5.508 install.
- Windows 2000 Professional (or XP Professional, if you must).
- Any SQL backend you own or are familiar with, as long as it comes with an ADO or ODBC provider. If you don't have a SQL database, you can download free trial versions from most major vendors that are good for anywhere from 60 days (DB2) to 120 days (MS SQL Server 2000). (</cmag/v2/v2n6cheapsql.html>)
- Run everything on one machine for development so that you don't need to deal with database administrators, web masters, networks, etc. In other words, eliminate as many possible sources of conflict as you can right at the beginning so that you can concentrate on learning Clarion/ASP.

Following my own advice, for this Clarion/ASP review I am using Windows 2000, the IIS 5.0 that came with it, Microsoft SQL Server 2000 Developer Edition, and whatever version of ADO SQL Server installed. I'll also be testing with Sybase Anywhere (and possibly TPS files) a little later on in this series.

## IIS and security

When you are setting up your machine for Clarion/ASP development, be sure to pay attention to the security settings in IIS. Many people aren't aware that IIS, FTP (file transfer) and SMTP (email) servers are all installed *and turned on* by default in, for example, Windows 2000 Professional. Unless you are behind a properly configured firewall (always a good idea) you are leaving yourself wide open to being compromised by any of a myriad of attacks that IIS is vulnerable to, especially if you have a constant connection like cable or DSL.

Apart from keeping up with the many patches for IIS, you should also download a copy of Microsoft's free [IIS Lockdown tool](#). In fact, I'd recommend it for any Windows 2000/XP user, as it makes it simple to disable or remove services that you don't actually need to have running, in addition to fixing a number of serious vulnerabilities.

## Installing the templates

Have you gotten the feeling by now that Clarion/ASP is not your usual Clarion add-in? Yup – I thought so. Maybe that's because it isn't – it's a whole new world. In any case, with all the rest of that stuff out of the way you are ready to actually begin installing and using the Clarion/ASP templates.

Physically, Clarion/ASP arrived in a small folder with a CD, an online training fact sheet and a registration card. The package did not include a printed manual or any "getting started" type of information.

I chose to use the custom install in setup just to see what the options would be. There were three choices: Examples, the Clarion/ASP templates and the Help/User manuals. No surprises there. The installer did want to add the Clarion/ASP icons to a new program group (Clarion 5.5) but I overrode that and stuck it in my C5.5 EE group (I need another program group on my Start menu like I need a 286.)

As part of my testing now I also uninstall products to see how cleanly they can be removed from my system. The Clarion/ASP uninstall seemed to work okay, other than it leaving an empty \C55\Examples folder (note – a long time ago I had removed this folder to free up disk space). This proved to be a problem, though, as I could not re-install the templates – I got over a hundred error messages complaining that it could not create various folders under that folder, nor could I access that folder from Windows Explorer (I kept getting 'Access Denied')



messages.) It took a full shutdown/restart in Windows 2000 in order to get the folder to disappear so that I could re-install and proceed.

Other than my fun with uninstalling the templates, it was a nice, boring, no surprises install – just the way I like them.

## Final install thoughts

Although both of the components I've discussed so far installed without problems, I'd like to see SoftVelocity make the available installation options much more obvious to the user, i.e., a front-end menu for the disk that lists all of the installable options and lets you easily choose which ones you'd like to run.

For example, because of an email mix-up (more on this next week), I didn't receive an email that talked about installing the SoftVelocity Discussions client. The disk itself auto-starts the install for the templates, and so I assumed that that was all there was on the CD. Once I found out about the Discussions client, I poked around on the disk a bit more and also found an evaluation version of the [TopStyle Pro](#) v2.5 Cascading Style Sheet editor from Bradbury Software. Neither TopStyle nor the Discussions client software were mentioned anywhere on the disk itself.

While I'm grouching, I'd also like to see some kind of printed getting started document included with the disk – even a single page would help in getting the new user oriented on how to begin. Relying on email is just too chancy, in my opinion.

That takes care of installation. [Next week](#) I'll look at the documentation, and I'll create my first Clarion/ASP web page.

---

*A longtime Clarion user, [Tom Hebenstreit](#) is an admitted tool junkie who refuses to go straight and code without his arsenal of third party products. During those rare moments when he isn't either using or writing about Clarion, he indulges his twin passions for blues and beer by performing around Southern California in a variety of totally-obscure-but-famous-any-day-now rock and blues bands.*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2002 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.

For marketing your Applications  
and Developer Accessories or to  
purchase other 3rd Party Tools ...

Developer **PLUS**™

[Home](#) [COL Archives](#)

[Topics](#) > [Tips/Techniques](#) > [Clarion Language](#)

## Data Structures And Algorithms Part V - The Sort Goes On

by **Alison Neal**

Published 2002-07-17

In my [last article](#) I introduced the Ordered Linked List and a variety of sort algorithms, including the Bubble, Cocktail Shaker, Insertion, and Counting sorts. I also explained that the Ordered List allows me to maintain data in order from the point of insertion, whereas sort algorithms allow me to sort the data after entry. In this article I will be continuing on this theme and will be covering the following algorithms:

- Radix Sort
- Selection Sort
- Heap Sort
- Quick Sort

I measured the efficiency of a sort algorithm in terms of clicks. A click represents 1/100<sup>th</sup> of a second. The sorts that I covered showed the following results, with 2,500 numbers ranged between 1 and 999:

Sort Name	Unsorted Array		Sorted Array	
	Clicks	Array Size	Clicks	Array Size
Bubble Sort	824	clicks	650	clicks
Cocktail Shaker	1526	clicks	1340	clicks
Insertion Sort	613	clicks	1	clicks
Counting Sort	0	clicks	0	clicks

The Counting Sort appears to be the quickest of the covered sort algorithms but has the added overhead of a temporary array, defined to the size of the number range. This means that the

Counting Sort algorithm is only truly efficient when sorting low numbers. The sort algorithms I'll cover next offer improved performance on large data sets as well.

## Radix Sort

The Radix Sort, like the Insertion Sort, is reasonably efficient where data is likely to be in near sorted order.

Sort Name	Unsorted Array		Sorted Array	
Radix Sort	1590	clicks	34	Clicks

This means that the Radix Sort and the Insertion Sort are good options if you have to re-sort data after only a few changes. Here's the code:

```

k = 10

LOOP WHILE k <= pMax * 10
  CLEAR(bucket)
  LOOP I = 1 TO pSZ
    LOOP j = 1 TO pSZ
      IF ~bucket[((pArray[i]%k)/(k/10))+1][j]
        bucket[((pArray[i]%k)/(k/10)+1)[j] = pArray[i]
        BREAK
      END
    END
  END
  CLEAR(pArray)
  l = 1
  LOOP i = 1 TO 10
    LOOP j = 1 TO pSZ
      IF bucket[i][j]
        pArray[l] = bucket[i][j]
        l+=1
      ELSE
        BREAK
      END
    END
  END
  k *= 10
END

```

The Radix Sort sorts the data, by digit place, into buckets and then re-merges those buckets back into the original array. What I mean by digit place is that the Radix Sort starts with the right-most digit, the ones place, and places each element into a bucket according the one's place value. It then remerges the buckets into a single array, and moves to the ten's place, and so on. The radix sort is similar to the counting sort, except that as it is only looking at one digit at a time the algorithm only needs from 0 – 9 buckets (10) over which to spread the data.

22	99	4	12	33	9
22	12	33	4	99	9
4	9	12	22	33	99

So, from the example above the Radix Sort will sort the ones thus:

0	1	2	3	4	5	6	7	8	9
		22	33	4					99
		12							9

Merging the bins back in the following order:

22	12	33	4	99	9
----	----	----	---	----	---

Then, the Radix Sort will work on the 10's position, sorting the numbers into buckets thus:

0	1	2	3	4	5	6	7	8	9
4	12	22	33						99
9									

Then merging the bins back into their correct order:

4	9	12	22	33	99
---	---	----	----	----	----

## Selection Sort

The Selection Sort is substantially faster than the Radix Sort when sorting unsorted data, however it is slower than the Radix Sort when the data is in sorted or near sorted order..

Sort Name	Unsorted Array		Sorted Array	
	Selection Sort	344	clicks	376
Radix Sort	1590	clicks	34	Clicks

The code for the selection sort is:

```

LOOP i = 1 TO pSZ
  xMin = i
  LOOP j = i + 1 TO pSZ

```

```

    IF (pArray[j] < pArray[xMin])
        xMin = j;
    END
END
swap(pArray[i], pArray[xMin])
END

```

The Selection Sort looks for the lowest number and moves that number into the appropriate position. So, with the example, the sort begins this way:

22	99	4	12	33	9
----	----	---	----	----	---

The outer loop starts at position one, value 22, and the nested loop starts at position two, value 99, then on finding position three, value 4, the minimum number (`xMin`) is set to be position three thus marking the position of the lowest number. The nested loop then continues to the end of the array finding nothing lower than the value 4.

When the nested loop finishes, values 4 and 22 are swapped. The outer loop then starts with position two, value 99, and the nested loop marks value 9 as the next lowest number, swapping values 9 and 99 when finished.

The outer loop then starts with position three, value 22 (swapped with 4 in the first pass), and the nested loop identifies 12 as the next lowest number, swapping these two values once finished.

At this point the array happens to be in the correct sort order, but the process continues until all the coded loops are completed.

22	99	4	12	33	9
4	99	22	12	33	9
4	9	22	12	33	99
4	9	12	22	33	99

## Heap Sort

In my testing I found the Heap Sort faster than both the Selection Sort and the Radix Sort, even though it takes the most steps.

Sort Name	Unsorted Array		Sorted Array	
	Heap Sort	30	clicks	32

Selection Sort	344	clicks	376	clicks
Radix Sort	1590	clicks	34	clicks

The Heap Sort differs from the Selection Sort in that its priority is the highest number not the lowest; that is, it sorts the array from right to left, rather than from left to right. At first I found this to be counterintuitive. There are two main loops contained in the Heap Sort. The purpose of the first loop is to move the highest number into the first position, which is what Heapdown does.

```

LOOP i = pSZ/2 TO 1 BY -1
  HeapDown(pArray,i,pSZ)
END

```

The purpose of the second loop can be summarised in pseudocode this way:

```

While Size > 1
  Swap(1,Size)
  Decrement(Size)
  Walk the (new) first element down into the (shorter) heap.

```

The code for the second loop is:

```

LOOP i = pSZ TO 2 BY - 1
  swap(pArray[1],pArray[i])
  HeapDown(pArray,1,i-1)
END

```

The HeapDown procedure code looks like this:

```

HeapDown      PROCEDURE(*ULONG[] pArray, ULONG pRoot, ULONG pBottom)
done = 0;
LOOP WHILE ((pRoot*2 <= pBottom) AND (~done))

  IF (pRoot*2 = pBottom)
    maxChild = pRoot * 2
  ELSIF (pArray[pRoot * 2] > pArray[pRoot * 2 + 1])
    maxChild = pRoot * 2
  ELSE
    maxChild = pRoot * 2 + 1
  END
  IF (pArray[pRoot] < pArray[maxChild])
    swap(pArray[pRoot],pArray[maxChild])
    pRoot = maxChild
  ELSE
    done = 1
  END
END

```

Going back to the first loop in the Heap Sort algorithm, and remembering that the purpose of



this loop is to get the highest number into the first position, you can see that the sort code moves the data as follows:

22	99	4	12	33	9
22	99	9	12	33	4
99	22	9	12	33	4
99	33	9	12	22	4

At the very start of the first loop `pSZ` is 6 (the size of the Array), so the first call to `HeapDown` is made with a `pRoot` value of 3 ( $pSZ / 2$ ) and a `pBottom` value of 6.

In `Heapdown` the loop goes through a single iteration setting `maxChild` to 6. As position three, value 4, is less than position six, value 9, the two values are swapped.

In the next call to `HeapDown` `pRoot` is 2 and `pBottom` is 6. The `Else` statement is triggered and `maxChild` is set to 5; no swaps are performed because 99 is higher than 33.

`HeapDown` is called again with `pRoot` set to 1 and `pBottom` set to 6. This time the second `IF` statement is triggered; `MaxChild` is set to 2, and values 99 and 22 are swapped putting the highest value of the array into the first position.

The `HeapDown` loop iterates again as `pRoot` is now only 2 and even multiplied by 2 is still not greater than `pBottom`. This time the `ELSE` statement executes setting `MaxChild` to 5, and another swap is performed between 33 and 22. `pRoot` is now set to 5 and thus the loop terminates returning to the original calling procedure, as  $5 * 2$  is  $> 6$ .

The first loop of the calling procedure also terminates. The second loop then makes the following changes:

4	33	9	12	22	99
33	4	9	12	22	99
33	22	9	12	4	99
4	22	9	12	33	99
22	4	9	12	33	99
22	12	9	4	33	99
4	12	9	22	33	99
12	4	9	22	33	99
9	4	12	22	33	99
4	9	12	22	33	99

Values 4 and 99 are immediately swapped, putting the value 99 in its correct position. A call is made to the HeapDown procedure, with pRoot being 1 and pBottom being 5.

In the first iteration of HeapDown the second IF statement executes, making maxChild equal to 2, values 33 and 4 are swapped, and pRoot now equals 2. In the second iteration the ELSE statement executes and maxChild equals 5, as value 4 is less than 22 they are swapped, and pRoot is made to equal 5, thus terminating the loop.

In the loop in the calling procedure I is now equal to 5, so positions one, value 33, and five, value 4, are swapped. Note that value 33 is now in the correct position and is not moved again. HeapDown is called again with pRoot being 1 and pBottom being 4. In the first iteration the second IF statement executes setting maxChild to 2, values 22 and 4 are swapped, and pRoot equals 2. In the second iteration, the first IF statement executes setting maxChild to 4, values 4 and 12 are swapped and pRoot is made to equal 4, thus terminating the loop.

Returning to the calling loop, I is now equal to 4 so positions one, value 22 and four, value 4 are now swapped. Now 22 is in the correct position and will not be moved again. HeapDown is called again with pRoot being equal to 1 and pBottom being equal to 3. The second IF statement executes making maxChild equal to 2, values 4 and 12 are swapped, and pRoot is set to 2 effectively terminating the loop.

Again, returning to the calling loop, this time I is now equal to 3 and so positions one, value 12, and three, value 9, are now swapped. This puts value 12 in its final resting position. HeapDown is called once more with pRoot being equal to 1 and pBottom being equal to 2. This time the first IF statement executes setting maxChild to 2, however 9 is not less than 4, so the loop is terminated without making any changes.

Then on returning to the calling loop, I is now equal to 2 and so position one, value 9 and two, value 4 are now swapped. So, the array is now in the correct order. However, Heapdown is called one more time with pRoot being equal to 1 and pBottom being equal to 1, so the loop never iterates.

## Quick Sort

The next quickest algorithm under my test conditions of 2,500 numbers between 1 and 999 was the Quick Sort

Sort Name	Unsorted Array		Sorted Array	
Quick Sort	22	Clicks	1053	clicks

Heap Sort	30	Clicks	32	clicks
Selection Sort	344	Clicks	376	clicks
Radix Sort	1590	Clicks	34	clicks

Note however that the quick sort can be quite slow where data is in sorted or near sorted order. There are, however, ways of improving on the following code:

```

l_hold = pLeft
r_hold = pRight
pivot = pArray[pLeft]
LOOP WHILE (pLeft < pRight)
  LOOP WHILE ((pArray[pRight] >= pivot) AND (pLeft < pRight))
    pRight-=1
  END

  IF (pLeft <> pRight)

    pArray[pLeft] = pArray[pRight]
    pLeft+=1
  END

  LOOP WHILE ((pArray[pLeft] <= pivot) AND (pLeft < pRight))
    pLeft+=1
  END

  IF (pLeft <> pRight)

    pArray[pRight] = pArray[pLeft]
    pRight -= 1
  END
END
pArray[pLeft] = pivot
pivot = pLeft
pLeft = l_hold
pRight = r_hold
IF (pLeft < pivot) THEN quickSort(pArray, pLeft, pivot-1).
IF (pRight > pivot) THEN quickSort(pArray, pivot+1, pRight)

```

The Quick Sort uses a divide and conquer strategy, selecting a datum (in this instance the first value) as a *pivot*, and moving this into its correct position by placing lesser values to the left and higher values to the right. However if the array is already in sorted order then the pivot is also already in the correct position, and the algorithm is slowed down by the continuous left to right scans that it then performs. To fix this problem it is possible to take the pivot from the middle position of the array.

Let's take a look at the example I've used with all the other sort algorithms:

22	99	4	12	33	9
9	12	4	22	33	99

4	9	12	22	33	99
---	---	----	----	----	----

At the beginning `pLeft` is equal to 1, and `pRight` is equal to 6 (size of the array) so `l_hold` equals 1, `r_hold` equals 6, and `pivot` equals 22 (the first value in the array). The first loop iterates as 1 is less than 6. The first nested loop never iterates as position six, value 9, is less than 22. The `IF` statement then executes as 1 does not equal 6 so position one now becomes 22, effectively giving me:

9	99	4	12	33	9
---	----	---	----	----	---

Remember that while position one has been overwritten the value is still stored in the `pivot` variable.

`pLeft` is then incremented by 1. The second nested loop does not iterate as position two, value 99, is greater than the pivot value of 22. The second `IF` statement then executes as `pLeft` does not equal `pRight`. So, position six is overwritten with the value 99 and `pRight` is decremented to 5.

9	99	4	12	33	99
---	----	---	----	----	----

The outer loop then goes through another iteration as 2 is still less than 5. The first nested loop then iterates once as position 5, value 33 is greater than the `pivot` value of 22, but position 4, value 12 is not greater than 22.

The first nested `IF` statement then executes as 4 does not equal 2. Position 2, value 99 is then overwritten with the value 12.

9	12	4	12	33	99
---	----	---	----	----	----

The second nested loop then iterates twice as position two, value 12, is less than 22, and position three, value 4 is also less than 22, however now `pLeft` and `pRight` are both 4. This effectively terminates the outer loop.

At this point the pivot value is returned to its correct position in the array.

9	12	4	22	33	99
---	----	---	----	----	----

The pivot is set to 4, and `pLeft` and `pRight` are restored to their original values of 1 and 6 respectively. As `pLeft` is less than the pivot point the first recursion occurs.

In this recursion `pLeft` is 1, and `pRight` is 3 (pivot point minus one), and the pivot value is set

to equal position one, which is 9. The outer loop iterates as 1 is less than 3, however the first nested loop never iterates as position three, which is 4, is less than the pivot value of 9.

The first IF statement then executes as 1 still does not equal 3. So, position one is made to equal 4.

4	12	4	22	33	99
---	----	---	----	----	----

pLeft is then incremented to equal 2. The second nested loop then iterates as position two, value 12 is not less than 9, and the second IF statement executes as 2 still does not equal 3. This time position three is made to equal 12 and pRight is decremented to be 2, effectively terminating the outer loop.

4	12	12	22	33	99
---	----	----	----	----	----

At this point the pivot value is restored in the array in it's correct position.

4	9	12	22	33	99
---	---	----	----	----	----

So, the array is now sorted correctly, but the algorithm will continue. The pivot is set to 2, pLeft and pRight are restored to their original values of 1 and 3 respectively, and the first IF statement calling the recursive function executes. However this time as pLeft and pRight both equal 1 the main loop never iterates and neither do the recursive IF calls.

Returning to the calling function, where pLeft is 1 and pRight is 3 the second IF statement then executes as 3 is still greater than the pivot point of 2. This time the procedure is called with pLeft of 3 (pivot point plus one) and a pRight of 3, so again nothing actually happens with this call. This effectively terminates the calling procedure, and returns to the very first calling function, where the second IF statement executes again; pRight was 6 and the pivot point was 4, so in this recursion pLeft is 5 and pRight is 6. This time the loop does execute, although it doesn't actually do anything and at the end both pLeft and pRight equal 6. As pLeft is then restored to a value of 5 and the pivot is 6, the process recurses yet again.

The fact that processing (recursive calls) continues after the array has been already sorted can really slow the Quick Sort algorithm down. You can see this by the large number of clicks taken to sort an already sorted array.

In the [next installment](#) I'll cover the Merge Sort and the Shell Sort algorithms, which do not suffer the same problems as the Quick Sort.

## [Download the source](#)

---

[Alison Neal](#) has been using Clarion since 2000, whilst working for [Asset Information Systems](#) (AIS) in Auckland, New Zealand. Some years ago (at the tender age of 19) Alison graduated from the Central Institute of Technology in Wellington, New Zealand with a major in Cobol. She also has a BA in English literature and has studied Computer Science, Philosophy and Information Systems. AIS is an independent division of Asset Forestry Ltd, and has a team of five programmers developing almost exclusively in Clarion. AIS also offers web (ClarioNET) and email services for the customer who needs everything. The company has many and varied customers bridging across a wide range of industries including Telecommunications, Forestry & Agriculture, Manufacturers, Military & Government, Legal & Financial, and Retail.

### Reader Comments

[Add a comment](#)

**[One note on the Quick sort. Using the insertion sort when...](#)**

Copyright © 1999-2002 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.

**INVEST**  
in your own abilities**Clarion**  
magazine[Home](#) [COL Archives](#)[Topics](#) > [News](#) > [ClarionMag 2001 News](#)

## **Clarion News**

Published 2001-11-21

### **INN Bio & News for 29-Jul-2002**

This week, the Ictips News Network is pleased to present another South African... or would that be a Floridian? Originally a chemist, now an independent software developer, she has worked in some really interesting places and has some good stories... not to mention a rather unusual (but really cool) wedding photo. Also on the site, Sue's new "Marketing Moments" series, and some SQL news.

*Posted Tuesday, July 30, 2002*

### **Server-Side PDF Generation**

Tracker Software has completed work on the first release of server-side PDF generation from Clarion Reports (client-side has been available for a while). As always there are Legacy and ABC examples, along with revised documentation to accompany the new functionality. This product offers support for your favorite reporting tools in addition to the Clarion Report's Templates, such as CPCS, Fomin and RPM and there is no reason why others should not function also. This has so far been tested (examples provided) with C55 and both ClarionNet and Web Builder. Please note: No work or testing has taken place with earlier Clarion versions.

*Posted Tuesday, July 30, 2002*

### **xTransparent Window v1.1**

Now available from SealSoft, xTransparent Window 1.1 is a class and extension Template that adds transparency to a Clarion window. All parameters can be set via variables. FadeIn and FadeOut effects are supported. xTransparent only works on non-MDI windows. Demo available.

*Posted Tuesday, July 30, 2002*

### **Dalby Source Printer Forum**



If you want to share opinions, receive help, recommend new features or simply get in contact with other Dalby Source Printer users, you can do so in the Dalby Source Printer Forum, which is a online discussion group dedicated to Dalby Source Printer users.

*Posted Tuesday, July 30, 2002*

### [ClarionNET Deployment Manager Available At ClarionShop](#)

The ClarionNET Deployment Manager, written by Michael D. Brooks is now available from ClarionShop. Just click on the ClarionNET logo in the ClarionShop product listing.

*Posted Tuesday, July 30, 2002*

### [New Serial Communications Library](#)

Andrey Krivoshein has released an object library which allows you to develop Clarion serial communication applications for 32-bit Windows. This library contains a SerialEvent class, which is an event-driven communication component. Cost is \$15.

*Posted Tuesday, July 30, 2002*

### [gFileFind 1.6 - Free Gift With Purchase](#)

Gitano's gFileFind 1.6 is now available. This is a free upgrade for registered users. New in this build: Local compile now possible; Added new 'Sort' fields, which preserves the file name for case sensitive instances; The example application has been greatly enhanced with record tagging, popup menus, and file operations. Buy before August 1st and you get a gift, your choice of IconsXP or PowerSearch. Even if you decide to return gFileFind, you keep your gift.

*Posted Tuesday, July 30, 2002*

### [Clarion/ASP 1.1 Available](#)

Clarion/ASP 1.1 is now available as a free update to purchasers of 1.0. New features include: At runtime Clarion/ASP can now include external HTML files into the document, allowing for easier management of site wide "boiler plate" text and graphics, additional incorporation of dynamic contents, and compound documents; ASP Source Code Procedures which allow you to execute scripts between pages; Filtered Selects (example: the end user fills in a state code, and the filtered select browse lists only items in that state); More embed points (59 in total); more opportunities to override such items as Select and Delete statements.

*Posted Monday, July 29, 2002*

### [Nextage Security Templates](#)

NextAge has released its Security Templates, implementing Category and Access Level based security with 9 levels and unlimited categories. For example, a user can be a level 1 for Accounts Payable and a level 9 for General Ledger. Developer sets default values, but all level requirements are configurable by end users. Features include: Add security to entire app with one extension; Set security parameters globally; Customize individual procedure security as needed; Add additional security checking to individual embeds (i.e. event:accepted for a

button); Jump start your applications security with simple Logon Window; Require different security levels for Add/Change/Delete on EIP browses and Forms; Enforce security with a simple User level paradigm or use Categories to setup authorizations based on the users group. Pricing is \$99.95 (US) which includes the templates and documentation in Acrobat form. Demo available.

*Posted Friday, July 26, 2002*

### **[Screen Capture program](#)**

Peck's screen capture program is now available from ShareIt! Numerous features include: Capture screen; Capture window; Capture area; Acquire from scanner; Print image; many image processing and enhancement capabilities; color conversions and enhancements. For a limited time The product costs US\$15 for a limited time.

*Posted Friday, July 26, 2002*

### **[Dalby Source Printer Version 5.1](#)**

Version 5.1 of the Dalby Source Printer removes the unnecessary creation of a TESTOUT.DSP file when printing.

*Posted Friday, July 26, 2002*

### **[KSMsgBox - A Thread Safe Message Box](#)**

KSMsgBox from Klarisoft is a thread safe message box class which replaces the standard Clarion MESSAGE dialog box. You continue using the MESSAGE function - no changes to your code are needed, except one line of code in the app setup. Demo available.

*Posted Friday, July 26, 2002*

### **[Firebird 1.5 Alpha](#)**

Firebird 1.5 is expected to be in alpha release in August 2002. Firebird is the open source version of Borland's Interbase SQL server.

*Posted Friday, July 26, 2002*

### **[ImageEx 1.1 Released](#)**

Version 1.1 of ImageEx is now available. This is a free update for all registered users. Changes include: Added BLOB support to the extended image and viewer controls (you can now store all additional image types like PNG, TIFF, PCD, ... to BLOBs in their native(!) formats); Improved performance of the extended image control (more speed, less memory usage); Added rotate / flip functions to the viewer control as well as grayscale and color inversion; Added CPCS report support to the extended image control. Demo available.

*Posted Friday, July 26, 2002*

### **[Duke Application Shell Bundle/Pricing Changes](#)**

Effective August 1, 2002 Gerald W Duke and Company has bundled and re-priced the

Application Shells to make them more affordable. Also effective August 1, 2002 Office Plus and QwikChex shells will no longer be available, although support for existing customers will continue. The Application Shell Basic Edition (General Ledger, Accounts Payable, Accounts Receivable, and Check Reconciliation) is \$449.95US. The Application Shell Popular Edition (General Ledger, Accounts Payable, Payroll, Check Reconciliation, Order Entry, Accounts Receivable, Inventory, and Purchase Orders) is \$749.95US. The Application Shell Enterprise Edition (General Ledger, Accounts Payable, Payroll, Check Reconciliation, Order Entry, Accounts Receivable, Inventory, Purchase Orders, Point of Sale, Contact Manager, Appointment Scheduling, List Manager, Job Cost Reporting, and Expense Report Tracking) is \$1149.95US.

*Posted Friday, July 26, 2002*

### [Icetips Takes Over CCS SQL Templates](#)

Andy Stapleton has announced that effective Monday, July 22 2002, the CCS SQL Templates will be sold, maintained and supported by Icetips Software, in partnership with Cowboy Computing Solutions. Andy cited overwhelming commitments with his new businesses (Dbwired and Paywire) as the reason for turning over the reigns to Icetips. Andy will be a technical consultant on the templates, and will continue to use them for his own desktop development.

*Posted Friday, July 19, 2002*

### [SealSoft Update](#)

New versions of xSearch, xQuickFilter, and xDataBackup are available. xSearch v1.1 changes: Support for Fomin Runtime Report. xQuickFilter 2.12 changes: Support for Fomin Runtime Report; Wildcard filter; Automatically remove prefix from field name when xQuickFilter popup menu is generated. xDataBackup Manager Pro v1.3 changes: Small bufix re error on simultaneous open configuration file by several workstations.

*Posted Wednesday, July 17, 2002*

### [LinderSoft News For July 2002 - Issue 1](#)

LinderSoft's first newsletter is now available. This is the easy way to keep up with the latest news about Linder products and technology. It will also inform you of changes to website, what's been happening at the Linder Software office and any discounts promotions currently running. Issue 1 includes a success story written by Robert Zauhere.

*Posted Wednesday, July 17, 2002*

### [New Lower SysAni Price](#)

SysAni, the animation control from solid software, is available now for \$29. This new lower price is not time-limited. If you're an existing customer of SysAni and you paid more than \$29, solid software is offering a \$10 discount on the purchase of any other solid software products. This discount *is* a time-limited offer, and expires at the end of July, 2002. When ordering at

ClarionShop, please put a note in the comments field saying that you're an existing SysAni user.

*Posted Wednesday, July 17, 2002*

### **[ABC Training With Bruce Johnson](#)**

CapeSoft's Bruce Johnson will conduct a two day workshop in Kansas on August 21 and 22, 2002. Please note that space is limited so book early to avoid disappointment. The title of the workshop is "Programming in Clarion's ABC" and will work through material in the book of the same name. The cost is \$240, which includes a copy of the book, lunches on both days, and the workshop itself. Registrants who already own a copy of the book get a \$40 discount.

*Posted Wednesday, July 17, 2002*

### **[Even More Free Software](#)**

Bob Healy reports that the free offer of the Home and Office Inventory program was so successful that this application is now available at no charge to anyone in the Clarion community. As a spin off to the Inventory program there is now a new Photo Album program based on the same technology. The Matrix Photo Album is a cute little app that allows you to maintain and group photos from vacations, birthdays, graduations and any event. Until the end of July anyone in the Clarion community can get a registered copy just for the asking and a download. The big one Bob is now looking for input on is Matrix's new church management system, The Stewardship Navigator. This program is derived from Matrix's popular and successful Non Profit management system Navigator Suite. The Stewardship Navigator is a full featured program to manage church memberships, contributions, pledges, attendance, family relationships, internal organizations and committees, accounting reports and statements and much more. For anyone providing input or suggestions on this program Matrix will provide your church (or any church of your choosing) a free registered copy of the program. For this program Matrix is looking for as much feedback as possible and would like to get it into as many churches as possible for the best field test. This offer is good through August 15, 2002. If you don't have a church that needs it right now Matrix will hold any qualifying registrations for you for up to six months. To download any of these programs and take advantage of the free software offers go to and select any of the above products for evaluation. For registration information please send your name and any program information to [sales@softwarebymatrix.com](mailto:sales@softwarebymatrix.com).

*Posted Wednesday, July 17, 2002*

### **[INN Bio And News For 15-Jul-2002](#)**

This week, the Ictips News Network is featuring its first Australian bio. Known on the newsgroups as... well, you'll see, he was an early joiner of the lazy programmer's club. And before that? He's another... you guessed it... musician! And he sent some great pics, too. Also on Ictips, the next installment of Sue and Arnor's adventure in Iceland, and some wonderful family news.

*Posted Wednesday, July 17, 2002*

### **[Gitano 60 day Money Back Guarantee](#)**

Gitano Software is offering a 60 day money back guarantee and free upgrades for a year on all Gitano products, all of which are now available from Gitano Software, ClarionShop, and DeveloperPlus.

*Posted Wednesday, July 17, 2002*

### **[EasyAutoEntry 1.00 Released](#)**

IngoSoftPlus's EasyAutoEntry class and template add field autocompletion to Clarion applications. Just add the template to the application, and all controls such as ENTRY, COMBO and SPIN (matching the given masks) will gain the AutoComplete property. (Edit-In-Place is supported only for the ABC templates.) The template lets you fine tune each field of each window and to group fields according to picture mask (in a mask you may use metasympols "\*" and "?"). Styles include: Suggested Part - expected continuation of a string automatically occurs as a selected part; PullDown List - all variants of expected continuation of a string occur in the pulldown list; Combined - if there is more than one variant of expected continuation of a string the PullDown List style is applied, and variants reduce as the user types characters, until there is only one variant left. Requires Clarion 5.0b or Clarion 5.5, 32 bit only. Demo available.

*Posted Wednesday, July 17, 2002*

### **[O-Mezz Update](#)**

A bug fix to O-Mezz (regarding SQL batch updates) is now available. TPS performance has also been improved. Developers who have edited the .TRN please make a backup. A new demo will be available next week.

*Posted Friday, July 12, 2002*

### **[245 Products At ClarionShop.com](#)**

New products at the recently redesigned ClarionShop.com include solid software's ImageEx, KlariSoft's ExcelWrite Class and PDFWrite Class, SealSoft's xAccess Manager Lite, IngaSoftPlus's chSTD function library, and of course Bruce Johnson's "Programming in Clarion's ABC" book.

*Posted Friday, July 12, 2002*

### **[gFileFind Money Back Guarantee](#)**

Gitano Software is offering a 30 day money back guarantee on its gFileFind product. Features include: Search for files in any drive folder or combination of them; Search for text in those files using AND & OR for complex searching; Preview images; Copy, move, delete, and launch the file from the browse; Sort by name, date, type, size, path, ascending, descending;

Filtering by date range, size range, hidden folder, system folders, etc.; Output search results to screen, file, or printer. gFileFind is \$79.95.

*Posted Friday, July 12, 2002*

### **[Clarion/ASP Introductory Offer Extended To July 31](#)**

Due to a high level of customer interest in the product and much positive feedback from the customers already taking the online training course, SoftVelocity has extended the Clarion/ASP special introductory offer. Purchase Clarion/ASP at the regular price (\$595) and get free: Clarion/ASP in 14 days - Online Level I training course (\$125 value); Six months Clarion/ASP Online Support Package (a \$150 value). This offer will expire Wednesday July 31. The Level I course includes six months access to text, labs, almost two hours of video and audio, and transcripts of all live classroom session chats so far (with more to come).

*Posted Wednesday, July 10, 2002*

### **[New ClarionShop.com Web Site](#)**

The ClarionShop web site is sporting a new design, including among other things a "list by date" feature which allows you to view new and recently updated products. There is also a Programs section for utilities and applications.

*Posted Wednesday, July 10, 2002*

### **[ImageEx Released](#)**

ImageEx, a Clarion addon that enhances imaging features, is now available. ImageEx consists of: An image control class that allows displaying all supported file types (see below) in a standard Clarion image control, both on windows and reports; An image viewer control with zooming (5% min, 5000% max) and panning, both fast and flicker free (anti-aliasing is available for better image quality); A thumbnail creator class. All these components are surrounded by code templates, making them easy to use. Supported file types include: JPEG, TIFF, GIF, PNG, SGI, Autodesk, Truevision, ZSoft Paintbrush, Kodak Photo-CD, Portable pixel/gray map images, Dr. Halo, SGI Wavefront, Windows bitmaps, Photoshop, Paintshop Pro, icons and metafiles. ImageEx supports ABC and legacy template chains, local & standalone runtime libraries, 32bit only. The installer also contains documentation in compiled HTML format as well as example applications for legacy and ABC templates. The price of US\$99 includes free updates and free email support. Demo available.

*Posted Wednesday, July 10, 2002*

### **[Bruce Johnson ABC Training In KC, Missouri](#)**

Capesoft's Bruce Johnson will be presenting his "Programming in Clarion's ABC" class in Kansas City, Missouri. The class is being expanded to two days so there will be more time for questions and discussion. All attendees will receive a copy of Bruce's book. Bruce will also be offering Capesoft accessories at special prices. Details to follow. Missouri. The class is being expanded to two days so there will be more time for questions and discussion. All attendees



will receive a copy of Bruce's book. Bruce will also be offering Capesoft accessories at special prices.

*Posted Wednesday, July 10, 2002*

### **[Flexible SQL Tables](#)**

Dan Pressnell started a thread in the Products.Drivers newsgroup on using a standardized local table definition to retrieve data from a variety of different SQL tables. There's been a lot of discussion on this point, and a lot of good information presented.

*Posted Wednesday, July 10, 2002*

### **[The JaDu Dictionary Browser Returns](#)**

The original JaDu dictionary browser utility makes a comeback after six years

*Posted Wednesday, July 10, 2002*

### **[xAccess Manager Lite v1.3 Released](#)**

SealSoft's xAccess Manager Lite v1.3 is now available. Changes include: List of Access Objects in global template sorted by name; GetUserName and GetUserStatus methods added; small bugfix to correct header translation in list of assigned users. New install, demo, and documentation available.

*Posted Wednesday, July 10, 2002*

### **[Updated DataModeller](#)**

A new version of DataMODeller is now available. If you have difficulty downloading the file, contact Johan de Klerk at [ycm@mweb.co.za](mailto:ycm@mweb.co.za).

*Posted Friday, July 05, 2002*

### **[SoftVelocity Holiday Hours](#)**

Please be advised that SoftVelocity's business offices will be closed Thursday, July 4th and Friday, July 5th for the US Holiday. The offices will reopen Monday, July 8th, operating as per usual business hours.

*Posted Thursday, July 04, 2002*

### **[Free Inventory Software Updated](#)**

Robert Healy has released an updated version of his Matrix Inventory software, incorporating many of the suggested changes. This software is free to all who give it a try and provide some sort of input on its usage, look & feel, design, ocumentation, help or ideas on price. Please send any responses to [sales@softwarebymatrix.com](mailto:sales@softwarebymatrix.com)

*Posted Thursday, July 04, 2002*

### **[Taboga Software News](#)**

A maintenance release of the Taboga Barcode Library is now available. Changes include:



Support for having user defined barcode drawing on reports, to allow printing barcodes in CPCS group header/footers; Barcodes on hidden tabs; Bug fixes. There will be a free upgrade for all existing Taboga Barcode Library customers when Clarion 5.6 comes around. However, with that release (whenever that is), the Taboga Barcode Library will increase in price from \$79.00 to \$129.00.

*Posted Thursday, July 04, 2002*

### [ABCFree Templates And Tools 2.41](#)

Version 2.41 of ABCFree Templates And Tools is now available for download. Changes include: Added wildcard and soundex support to fAddToFilter function; Fixed "Find Next" button on BrowseCallvsFindFunction; Minor bug fixes.

*Posted Thursday, July 04, 2002*

### [Open Source Database Links](#)

Skipping Dot Net has a linkfest page for open source databases. Good links here to information on the key players: MySQL, PostgreSQL, Interbase, SAP DB, and a few others. The only one missing is the Firebird Interbase variant (firebird.sourceforge.net).

*Posted Tuesday, July 02, 2002*

### [Remote Backup Code](#)

Bob Campbell has posted some code for doing remote backups. Read more in this newsgroup message on the SoftVelocity news server.

*Posted Monday, July 01, 2002*

### [ABC Bugs Listed](#)

Bruce Johnson has begun listing bugs found in the ABC classes, along with workarounds.

*Posted Monday, July 01, 2002*

### [New dbZIP Templates](#)

Dave Beggs has released the dbZIP templates, which add basic Zip and Unzip functions to Clarion applications. The templates use the Bigspeed ZIP Dll which you also need to purchase from BigSpeedSoft. This is an inexpensive (\$65), royalty free, very fast (faster than winzip), small footprint (64k) DLL. The templates allow you to create a zip file or unzip one in two steps: 1) Add a global extension; 2) Add a control template. The global extension contains the prototypes for all the methods used in the DLL, so you can use it as a head start to write your own much more involved zip and unzip code. The template is US\$18, and is still I beta. The price for the gold release may be higher.

*Posted Monday, July 01, 2002*

### [ASP Video 2](#)

Part 2 of the Clarion/ASP video download - approx 9 MB.

*Posted Monday, July 01, 2002*

### [ASP Video 1](#)

Part 1 of the Clarion/ASP video download - approx 9 MB.

*Posted Monday, July 01, 2002*

### [Icetips Professional Wizard New Build](#)

A new build of our Icetips Professional Wizards is now available. You can download a fully functional demo as well.

*Posted Monday, July 01, 2002*

### [MySQL Resources](#)

Ron Schofield has compiled a number of tips for MySQL development.

*Posted Monday, July 01, 2002*

### [Updated ImageMan Imaging Templates](#)

NextAge Consulting has released Version 1.01 of the Imageman Templates. This is the template chain that uses the Data-Techniques Imageman OCXs.

*Posted Monday, July 01, 2002*

## **Reader Comments**

[Add a comment](#)

Copyright © 1999-2002 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

## [Clarion Magazine](#)

clarion magazine  
**Good help isn't that hard to find.**

**\$1.67** per  
issue

[Home](#) [COL Archives](#)

[Topics](#) > [Non-tech](#) > [About ClarionMag](#)

### **Clarion Magazine Goes To Double Opt-In Email List**

Published 2002-07-22

This week marks a change in how Clarion Magazine sends out automated emails. Formerly, if you signed up as a member or subscriber, and simply indicated you wanted to receive email from Clarion Magazine, we sent it. This has now changed.

If you indicate on the Clarion Magazine [new member form](#) or [user update form](#) (login required) that you want to receive emails, we now send you a confirmation email. That email contains a link, and you have to click on that link to begin receiving automated email (hence the term “double opt-in”).

There are four email lists currently available:

- Notification of membership/subscription due for renewal
- Weekly update of new articles and news items
- Special events/planned downtime notices/site changes not covered in the weekly update
- Third party product information (which we may, in the future, send on behalf of third party vendors – we don't share your email address with anyone)

The double-opt-in approach has several benefits. First, it ensures that no one can subscribe you to an email list without your knowledge or consent. Second, it weeds out invalid addresses and full mailboxes. If you sign up for one or more mailing lists and don't get any email, go back to the [user update form](#) and make sure you've provided a working email address. This happens a *lot*. We have hundreds of undeliverable email addresses for people who said they wanted to receive the weekly update.

If we already have a valid email address on file for you, you will by now have received an email with the subject “This could be the last email you get from Clarion Magazine.” That email contains instructions on adding yourself to the new double opt-in email lists.

If you didn't get the email, no worries. Just do **one of the following**:

- If you're **already a free member or paid subscriber** (past or present), please go to the [user update form](#) and update your user information. Be sure to correct your email address, and make your email list selections. When you complete the form the server will email you a confirmation link. If you've forgotten your id and/or password email [subscriptions@clarionmag.com](mailto:subscriptions@clarionmag.com).
- If you **don't yet have a free membership or paid subscription** (past or present) go to the [subscription information page](#) and follow the instructions.

If you have any questions, please email [subscriptions@clarionmag.com](mailto:subscriptions@clarionmag.com).

David Harms  
Publisher, Clarion Magazine

## Reader Comments

[Add a comment](#)

Copyright © 1999-2002 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.

**Reborn Free****CLARION**  
*online*[Home](#) [COL Archives](#)[Topics](#) > [Internet](#) > [ASP](#)

## Clarion/ASP Review Part 2

**by Tom Hebenstreit**

Published 2002-07-23

[Last week](#) I gave an overview of some of the terminology and technology behind SoftVelocity's Clarion/ASP product. This week I'll look at the product documentation, and create a simple Clarion/ASP web page.

### Documentation

Clarion/ASP comes with two manuals, both in Adobe Acrobat PDF format. The first is a 158 page Users Guide that includes basic background information, a getting started chapter, reference sections on the Clarion/ASP templates and embeds, and a section on using the Style Sheet Editor. The second manual is a 188 page Annotated Examples document. This one is more step-by-step and detail oriented, and walks you through creating four increasingly complex applications. It also includes sections on integrating Clarion/ASP pages with existing web sites, and an ASP code walkthrough that goes through typical generated browse and edit scripts line by line. Both manuals appear to be well written (at least what I've gotten through so far) and are formatted in the familiar style used by other SoftVelocity documentation.

Rounding out the documentation are a couple of text files – a readme for late-breaking news and a quick-start guide to generating your first application.

### Setting up for Online training

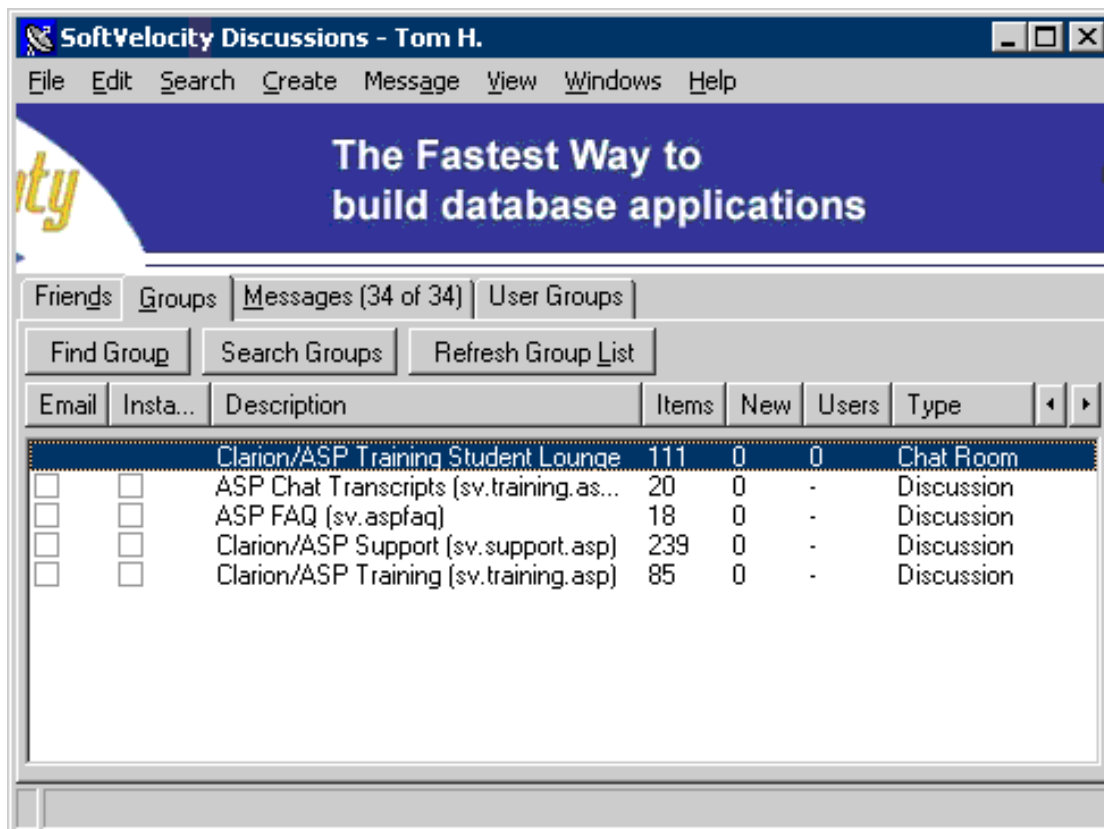
For a limited time, the Clarion/ASP online training course is being bundled with the Clarion/ASP templates – a wise decision on SoftVelocity's part, in my opinion. Like Windows programming and Clarion itself, Clarion/ASP is one of those paradoxical products that can be very simple to get started with, and yet it can take quite a while to become fully proficient. So, to get you over the initial hump, SoftVelocity has developed the first of what the company says

will be many online training courses for all of its products.

Clarion/ASP training has three basic components:

- Online, web-based HTML pages with text, graphics, audio and, in many cases, video segments. The various topics can be accessed anytime, and are available to go through or review at your own pace. Note: In order to view the online audio and video portions of the training, you must be using Microsoft Internet Explorer, and you must also have ActiveX turned on within the browser.
- Special dedicated newsgroups where class members can post questions. There is also a dedicated read-only FAQ (frequently asked questions) newsgroup where SoftVelocity posts answers to common questions.
- Interactive online chat. This includes scheduled sessions where the instructor goes over material live and handles questions right on the spot, the ability to hang out in a group chat room called the Student Lounge (think IRC) and also to talk individually with other Clarion/ASP users (ala ICQ, AIM, etc)

In order to facilitate the second and third items, SoftVelocity has developed a client application they call 'SoftVelocity Discussions'. It was provided on the Clarion/ASP disk, and installed without a hitch.



**Figure 1. The SoftVelocity Discussions main window**

My primary complaint with Discussions so far is that all of the help screens are provided online (i.e., across the Internet and not from the local machine), and for some reason they

simply do not work on my machine (no messages, no blank browser, just nothing at all happens when I click on the help menu items). I was under the impression that there wasn't any help until, during a chat, SV pointed me at the actual Discussions help URL. That worked fine directly from a browser, but not from within the program.

Note that while you can access the newsgroups via standard clients such as Virtual Access or Microsoft Outlook Express, I've found so far that using SoftVelocity Discussions lets me easily access both the groups and the chats, and has the additional advantage of keeping all class related information in one central, easily searched location. The SoftVelocity client also appears to have the ability to interact with ICQ and to do email, but I haven't tried those options yet. Additionally, the documentation states that SoftVelocity Discussions can also be accessed via a web browser (sigh, so many options to check out...)

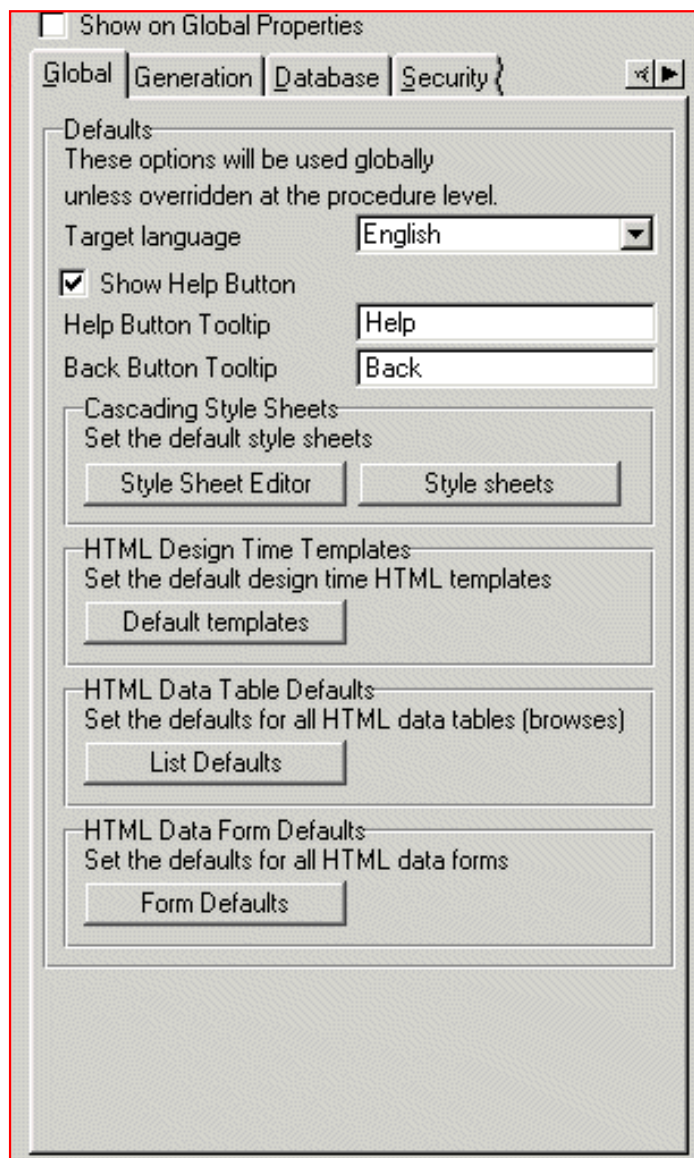
## **A Simple Start**

On the face of it, Clarion/ASP is pretty unimposing, containing only three templates: the global extension, a browse extension and a form extension. As you'll see this week, that means they can be used pretty easily. As you'll also see over the next two parts of this review, there are a lot of opportunities to get complex within those few templates.

Probably the most significant thing to keep in mind when using Clarion/ASP is that the contents of your app basically don't mean squat – the dictionary is king. All that an app really does is give you a place to hang the global extension and templates for any form and browse procedures that you want to generate ASP code for. Changes made to controls on the window, etc., in Window formatters have little to no effect on what Clarion/ASP generates (those types of changes are handled from within the Clarion/ASP templates). While this may be a bit dismaying at first glance, it is actually a pretty cool feature since it allows you to selectively add ASP templates to existing applications and yet keep the format of your Windows applications and their corresponding ASP pages independent. Each can be tailored to its proper environment.

To keep this simple, I'll apply Clarion/ASP to a small wizarded app (I'm assuming that most readers here already know how create a data dictionary and use the application wizard to create an app.) The next step is to apply the Clarion/ASP global extension as shown in Figure 2.

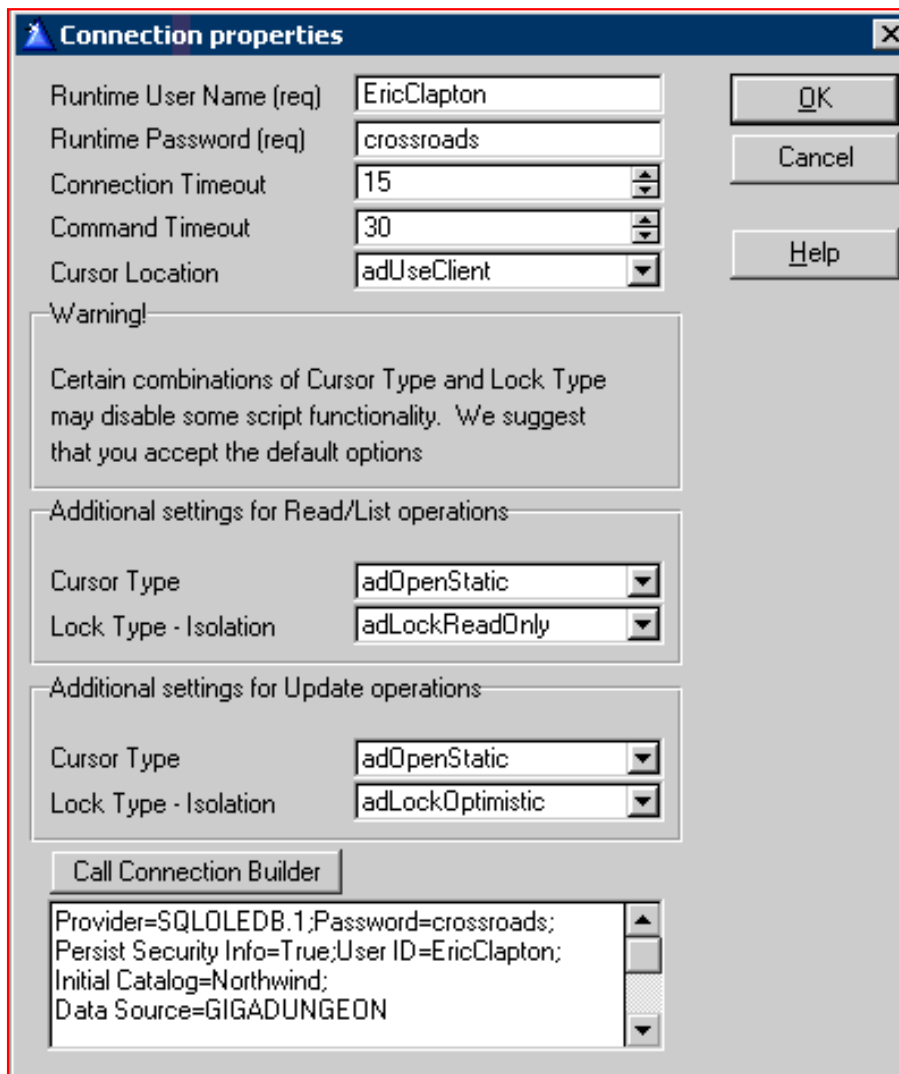




**Figure 2. The Clarion/ASP Global Extension main tab**

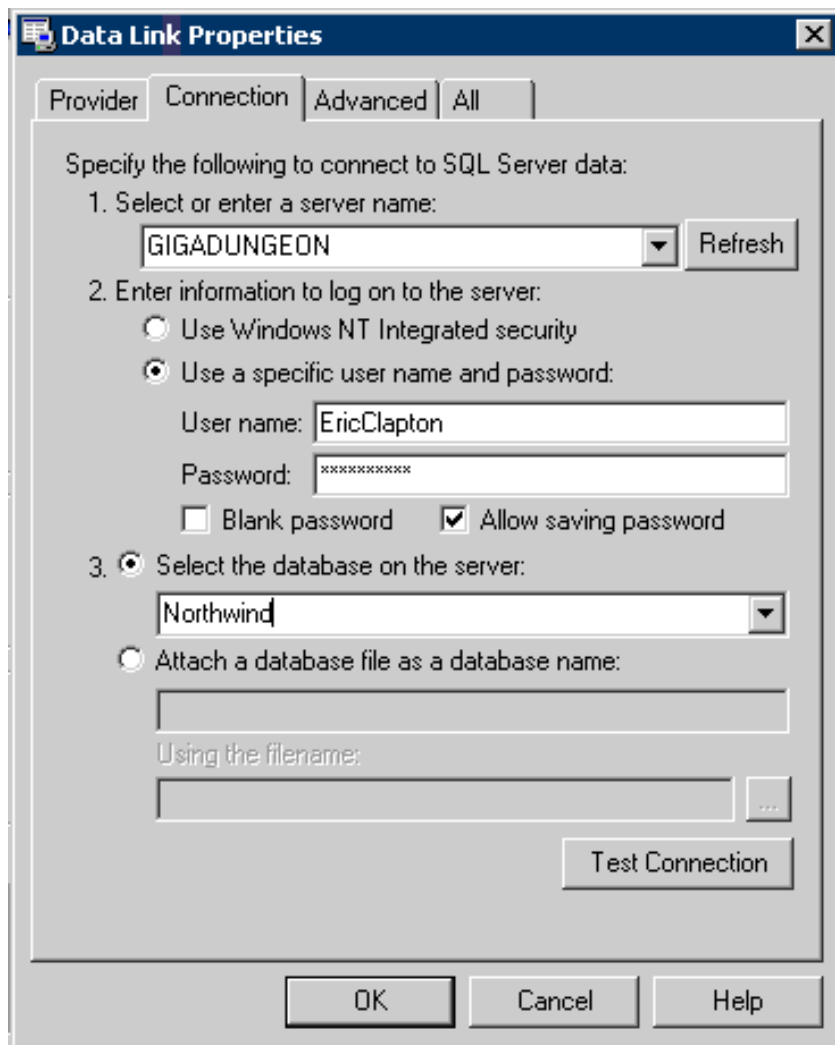
The global extension is where you can set the global defaults for your entire ASP application, and where you specify your database and its connection parameters. For now, I'll just do the one thing I have to do, and that is fill in the database information as shown in Figure 3 (I'll be covering some of the other global functions in the next part of this review.)





**Figure 3. Setting Clarion/ASP global database connection properties**

The connection properties tell the templates exactly how to connect to the database. The majority of the prompts on this screen are defaulted – the only things I filled in were runtime user name and password. The text box at the bottom was filled for me when I called the Connection Builder (see Figure 4).



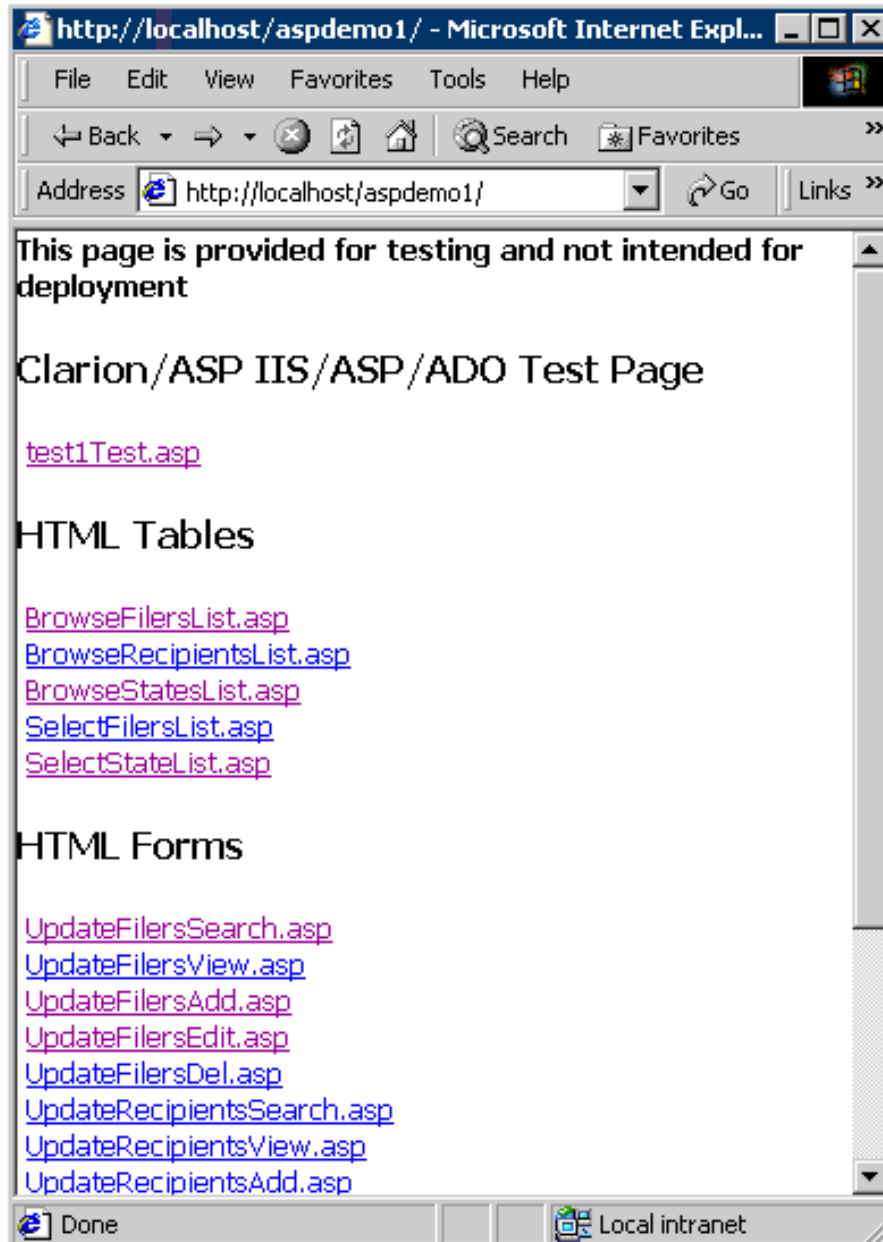
**Figure 4. The Connection Builder**

The Connection Builder makes it a snap to select your data provider from a list of all installed providers (I chose SQL Server), then specify the server, database, user ID and password to connect with. You can also test the connection right on the spot to verify that it works correctly. When you exit this dialog, the settings are automatically formatted and placed in the connection string box shown in Figure 3.

The next step is to save the global settings and add the Clarion/ASP browse extension to a browse procedure, then add the Form extension template to an update form and save that. Both templates will generate usable (if not pretty) code with the default settings.

To generate the Clarion/ASP web site, just click on Project, then Generate All. Note that you don't need to compile unless you want to, and you would only do that if you wanted to create the Clarion Windows app equivalent of the ASP code. The Clarion/ASP code generation is completely independent of the Clarion code and has no use for the compiler.

After copying the generated files to my IIS Clarion/ASP test site, I pointed Internet Explorer at the site and saw the display shown in Figure 5.



**Figure 5. Well, would you look at that -- it works!**

## Important Things I've Learned So Far...

Clarion/ASP is a new product, and that means there are a few wrinkles to be worked out, mostly in the documentation. Here are some of the things I've learned:

- Number one, be sure that before you purchase Clarion/ASP you have a valid SoftVelocity login for the email address you provide at the time of purchase. My SoftVelocity login email happened to be different than the one used for the purchase, so I did not receive crucial emails that contained the information on getting started with the online classes and about installing the Discussions client software. Because of this, I missed the first chat, couldn't access the online training and was generally confused about how to get started. It took a number

of emails to get it all straightened out, but to SoftVelocity's credit, they did respond quickly (like in ten minutes!) and got me fixed up with a minimum of hassle.

- Be sure you have scripting enabled in your web browser, or many (if not all) Clarion/ASP features will appear to be broken. Also, if you have a newer version of Norton AntiVirus, be sure that you set it to allow scripting in your test site or it will cause problems as well.
- In the Connection Builder, on the Data Link properties (see Figure 4), you must have 'allow saving password' checked or you will get very weird results when you run your application (all edits fail, adds fail even though they say success, numerous 'record not found' errors and so forth). I'd like to see this stressed even more in the online courses and in the manuals (obviously, I missed this at one point and it caused me no end of confusion and grief!)
- You'd better have your dictionary in tip-top shape for Clarion/ASP (this is good advice for any Clarion application.) Unlike your regular programs that take most of their information from the window structures stored in the APP file, Clarion/ASP takes little or no information from the application's windows – it all comes from the dictionary. So where you would normally be able to compensate for funky dictionary prompts, tooltips and so forth by fixing up the control properties directly on a window, that won't work in Clarion/ASP. If you don't change these properties in the dictionary you have to monkey with them via the templates, and that can get very tedious.
- It helps to add Index.htm (the Clarion/ASP default) to the list of default start files that IIS looks for when access your test site. For the lazy typist, this can save you a few keystrokes every time you access the site. You can do this on the Documents tab in the IIS Default Web Site properties.
- If you are nervous about having a web server running all the time on your machine, you can set IIS to be started manually. That way it is only running when you want it to be, and you can shut it down when you don't need it. In Windows 2000, click on Administrative Tools in the Control Panel, then Services. Scroll to the bottom and set the Startup Type for 'World Wide Web Publishing Services' to manual. You can then use the Internet Services Manager (also in Administrative Tools) to start and stop IIS as needed.
- My brain is full.

That's it for this installment – keep in mind that I've barely scratched the surface of what you can do with Clarion/ASP. See you next week!

---

*A longtime Clarion user, [Tom Hebenstreit](#) is an admitted tool junkie who refuses to go straight and code without his arsenal of third party products. During those rare moments when he isn't either using or writing about Clarion, he indulges his twin passions for blues and beer by performing around Southern California in a variety of totally-obscure-but-famous-any-day-now rock and blues bands.*

## Reader Comments

[Add a comment](#)

**Thanks for doing this!! I, and I am sure many others,...**  
**Tom I am following this every week and I have just gotten...**

Copyright © 1999-2002 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.

**Reborn Free****CLARION**  
*online*[Home](#) [COL Archives](#)[Topics](#) > [Tips/Techniques](#) > [Clarion Language](#)

## Custom Queue Sorting

**by Gordon Smith**

Published 2002-07-24

Clarion can sort queues. But it really only has two sorting algorithms, alphabetical and numeric. To be fair these are indeed popular options, but there are many examples when an alternative would be useful:

- When numerics and strings are mixed, or you have numerics stored as strings (the strings '1', '2', '10', '20' will normally sort alphabetically as '1', '10', '2', '20')
- When the sorted field contains weekdays or months.
- When the sorted field contains roman numerals

Deep within the bowels of the Clarion runtime there is indeed a way to provide your own sorting logic! It's a little, undocumented gem that looks like this:

```
map
  CompareProcedure procedure(*group l,*group r)|
    ,signed, type
  module('runtime')
    SortQ procedure(queue q, *CompareProcedure proc)|
      ,name('Cla$SORTqueuefkey')
  end
end
```

This procedure takes two parameters, the queue to be sorted, and the procedure which will be called to compare any two rows.

The SortQ procedure works in the same fashion as the standard Clarion SORT function (see [Alison Neal's articles](#) for more information on sorting), except when it comes to the actual comparison, it simply calls the nominated procedure to determine the actual precedent..

The compare procedure takes two parameters, `l` and `r`. The `l` parameter is the left hand side of the compare. It is a group structure which matches a single row in the queue being sorted. The `r` parameter is the right hand side of the compare. It is also a group structure which matches a single row in the queue being sorted.

A real-world declaration would look like this:

```

program
MyQueue      queue, type
stuff        string(256)
            end
CompareGroup group(MyQueue), type
            end

map
  DoCompare procedure(*CompareGroup l, *CompareGroup r) |
    , signed
  module('runtime')
    SortQ procedure(queue q, *DoCompare) |
      ,name('Cla$SORTqueuefkey')
    end
  end
theQueue MyQueue
code
  SortQ(theQueue, DoCompare)
DoCompare procedure(*CompareGroup l, *CompareGroup r)
code
  return choose(format(l.stuff, @n03) <
    format(r.stuff, @n03), -1, 1)

```

The above example (see the downloadable source) demonstrates how to sort a list of strings which contain numeric values correctly. The `DoCompare` procedure formats the strings with leading zeroes so they will compare properly as strings. To sort the queue, just call the `SortQ` procedure, passing in the queue and the label of your sorting procedure:

```
SortQ(theQueue, DoCompare)
```

`DoCompare` must return either `-1` or `1`, depending on how the left and right values compare. If you want the queue sorted in ascending order, return `-1` if the left is less than the right, and `1` if the left is greater than the right. For descending order use the reverse. You can also return `0` for exact matches (this will help optimize the sorting).

[Download the source](#)

---

*Prior to joining TopSpeed Development Centre, [Gordon Smith](#) worked for an Irish company developing software for multi-national pharmaceutical companies. He was also a member of the 3rd party accessories program (Compile Manager 2) and*

*developed the Clarion Class Browser. He is currently working with several of the original TopSpeed Developers at Seisint Inc..*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2002 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.



clarion magazine  
**Good help isn't that hard to find.**

**\$1.67** per  
 issue

[Home](#) [COL Archives](#)

[Topics](#) > [Tips/Techniques](#) > [Clarion Language](#)

## Data Structures And Algorithms Part VI - All Sorted Out

by **Alison Neal**

Published 2002-07-26

[Last week](#) I covered the Radix, Selection, Heap, and Quick sorts. I have only two more sorts to cover: the Merge sort, and my personal favorite, the Shell sort.

### Merge Sort

From the table below you can see that the Merge Sort is as quick with already sorted data as it is with unsorted data.

Sort Name	Unsorted Array		Sorted Array	
	Clicks	Time	Clicks	Time
Merge Sort	22	clicks	19	clicks
Quick Sort	22	clicks	1053	clicks
Heap Sort	30	clicks	32	clicks
Selection Sort	344	clicks	376	clicks
Radix Sort	1590	clicks	34	clicks

However the Merge Sort does still use recursion:

```

IF(first < last)
  mid = (first + last) / 2
  mergeSort(pArray, first, mid)
  mergeSort(pArray, mid+1, last)
  Merge(pArray, first, mid, last)
END

```

The Merge Sort splits the data into two groups, sorts each group, and then merges the groups back together.

22	99	4	12	33	9
22	99	4	12	33	9
4	22	99	12	33	9
4	22	99	12	33	9
4	22	99	9	12	33
4	9	12	22	33	99

Working through the same example, in the first call `first` is 1, `last` is 6 (the size of the array), and `mid` is therefore equal to 3. MergeSort is called again with `first` having a value of 1 and `last` a value of 3, making the `mid` value 2. MergeSort is called again with `first` having a value of 1 and `last` having value of 2, making `mid` 1. And MergeSort is called yet again with a `first` value of 1 and a `last` value of 1, meaning that the procedure returns without going any further.

The second recursive call is made from the calling function, with a `first` value of 2 and a `last` value of 2, which again causes the procedure to return without doing anything. The calls so far have been:

Step	Callr	Call	first	Last	mid
1			1	6	3
2	1	1	1	3	2
3	2	1	1	2	1
4	3	1	1	1	
5	3	2	2	2	

The third recursion is now going to call the Merge procedure with `first` being 1, `mid` being `last` and 1 being 2. Here's the code:

```

first1 = first
last1 = mid
first2 = mid+1
last2 = last
i = first1
LOOP WHILE((first1 <= last1) AND (first2 <= last2))
  IF(pArray[first1] < pArray[first2])
    bArray[i] = pArray[first1]
    first1+=1
  ELSE
    bArray[i] = pArray[first2]
    first2+=1
  END
  i+=1

```

```

END

LOOP WHILE (first1 <= last1)
  bArray[i] = pArray[first1]
  first1+=1
  i+=1
END
LOOP WHILE(first2 <= last2)
  bArray[i] = pArray[first2]
  first2+=1
  i+=1
END
LOOP i = F TO l
  pArray[i] = bArray[i]
END

```

Now the array is treated as though it were two separate halves, with `first1` (the beginning of the first half) being set to 1, `last1` (the end of the first half) being set to 1, `first2` (the beginning of the second half) being set to 2, and `last2` (the end of the second half) being set to 2. Note that each half at this stage only has one element in it.

The first loop executes as `first1` equals `last1` and `first2` equals `last2`. As 22 is definitely less than 99 the temporary array holds the lessor in first position, `i` is incremented by 1 and `first1` is incremented by 1 thus terminating the loop and causing the second loop to fail. The third loop executes as `first2` still equals `last2`, and as `xi` now equals 2, the value of position two is placed in position two of the temporary array, now giving 22,99. `first2` and `i` are both incremented and the loop is terminated. Then the temporary array is copied over the original, even though nothing has changed.

Returning to the calling procedure, the third recursion has now completely finished processing and returns to the second recursion which now continues with another recursive call:

Step	Caller	Call	F	l	mid
6	2	2	3	3	

As `first` and `last` both equal three nothing happens. The second recursion now calls the Merge procedure.

This time `first` is 1, `mid` is 2, and `last` is 3 making `first1` equal 1, `last1` equal 2, `first2` equal 3, and `last2` equal 3. The first half of the array has two elements and the second half of the array has 1. Effectively the algorithm is seeing this:

1st half		2nd half			
22	99	4	12	33	9

The first loop executes and as 22 is not less than 4, the ELSE statement executes, placing the value 4 into the temporary array in the first position. `i` is incremented to 2 and `first2` is incremented to 4.

`first1` is still less than `last1` so the second loop then iterates, and the temporary array becomes 4,22,99 before the loop terminates. The third loop never iterates, as `first2` was previously incremented to 4. The temporary array is then copied over the original:

4	22	99	12	33	9
---	----	----	----	----	---

The Merge procedure then returns to the calling MergeSort procedure, which in turn returns to its calling procedure as it has now done everything it was supposed to.

Step	Caller	Call	f	l	mid
7	1	2	4	6	5

Now the first recursive call makes its second call to MergeSort, this time passing the `first`, `last`, and `mid` with the values 4, 6, and 5 respectively. At this point you should be beginning to see that the Merge Sort algorithm first works down the left side and then down the right side of the array.

MergeSort then makes another call to itself, passing `first`, `last`, and `mid` as 4, 5, 4 respectively (step 8). This in turn makes another recursive call, which does nothing as `first` and `last` both equal 4 (step 9). It attempts to process the second recursive call, and again achieves nothing as `first` and `last` both equal 5 (step 10).

Step	Caller	Call	f	l	mid
8	7	1	4	5	4
9	8	1	4	4	
10	8	2	5	5	
11	7	2	6	6	

Now the Merge procedure is called again with the `first`, `mid`, `last` values of 4, 4, and 5 respectively. Setting `first1` to 4, `last1` to 4, `first2` to 5, `last2` to 5, and `i` to 4. The arrays the Merge procedure is seeing are as follows.

			1 <sup>st</sup> half	2 <sup>nd</sup> half	
--	--	--	-------------------------	-------------------------	--

4	22	99	12	33	9
---	----	----	----	----	---

As `first1` equals `last1` and `first2` equals `last2`, the first loop iterates. 12 is definitely lower than 33 so it is copied into the temporary array in the fourth position (the starting point). `first1` is then incremented so that it is now greater than `last1`, and so the loop terminates and the second loop never iterates.

As `first2` is still equal to `last2`, the third loop iterates; the value is copied into the temporary array. Then the temporary array is copied back over the original array, leaving the original array exactly where it was prior to the Merge procedure being called:

4	22	99	12	33	9
---	----	----	----	----	---

The MergeSort function has now completed everything that it was meant to so it returns, where its calling function now attempts its second recursive call (step 11). This call does nothing, however, as both `first` and `last` equal 6. On returning the Merge Sort function (step 7) then calls the Merge function once more with the `first`, `mid` and `last` being 4, 5 and 6 respectively. The array that the Merge procedure is looking at this time is this:

			1st half		2nd half
4	22	99	12	33	9

The first loop iterates and the ELSE statement executes, putting the value 9 in position four in the temporary array, and both `first2` and `i` are incremented. The second loop iterates with the value 12 being placed in position five and 33 being placed in position six of the temporary array. The temporary is then copied over the original:

4	22	99	9	12	33
---	----	----	---	----	----

Note that each half of the array is sorted within itself.

On returning to MergeSort all the recursive calls so far are completed, which just leaves the original MergeSort to call the Merge procedure with `first`, `mid`, and `last` having values 1, 3, and 6 respectively. Now the array looks like this:

1st half			2nd half		
4	22	99	9	12	33

`first1` equals 1, `last1` equals 3, `first2` equals 4, `last2` equals 6 and `i` equals 1. The

first loop iterates, and 4 is less than 9, so it takes the first position in the temporary array. `first1` and `i` are then both incremented to equal 2. This time around, 9 is less than 22 so the `ELSE` statement executes, copying 9 into the second position of the temporary array. `first2` is then incremented to 5 and `i` is incremented to 3. Now the comparison is performed between 12 and 22, with the value 12 being copied into the temporary array in the third position. The next iteration sees a comparison of 22 and 33 with 22 now being placed into the temporary array. Then 33 and 99 are compared, and 33 is copied into the temporary array. At this stage `first2` is incremented to 7, meaning that the loop is now terminated, but the value 99 still needs to be put into its correct position.

The second loop achieves this as `first1` currently equals 3 the same as `last1`, and so 99 is copied into the last position of the temporary array, giving:

4	9	12	22	33	99
---	---	----	----	----	----

On returning the calling procedure there are no more calls to be made. `MergeSort` finishes nicely without an excess of recursive calls continuing after the array has been sorted.

## The Shell Sort

The Shell Sort is my personal favorite. It is quick, without a large overhead, and it works well with data that is either in sorted or near sorted order.

Sort Name	Unsorted Array		Sorted Array	
	Clicks	Array Size	Clicks	Array Size
Shell Sort	18	clicks	6	clicks
Merge Sort	22	clicks	19	clicks
Quick Sort	22	clicks	1053	clicks
Heap Sort	30	clicks	32	clicks
Selection Sort	344	clicks	376	clicks
Radix Sort	1590	clicks	34	clicks

The Shell Sort uses a choice of numbers as a basis for the sort algorithm. x In my case I've used the following:

1,391,376
463,792

198,768
86,961
33,936
13,776
4,592
1,968
861
336
112
48
21
7
3
1

These numbers are used to repeatedly sort all items that are  $n$  positions apart, starting with a large value of  $n$  and decreasing  $n$  gradually. I've chosen mine arbitrarily, but you can also easily calculate some usable numbers:

$2^n - 1$  would give you 1, 3, 7, 15, 31....

$3^n - 2$  would give you 1, 7, 25, 79...

Here is the code:

```

LOOP k = 1 TO 16
  temp = incs[k]
  LOOP x = 1 + temp TO pSZ
    l = pArray[x]
    j = x
    LOOP WHILE j > temp AND pArray[j-temp] > l
      pArray[j] = pArray[j-temp]
      j -= temp
  
```



```

    END
    pArray[j] = 1
  END
END

```

The first loop is for the quantity of numbers that you are using as steps. I have only used 16 numbers so I only need to perform this loop 16 times.

22	99	4	12	33	9
12	99	4	22	33	9
12	33	4	22	99	9
4	12	33	22	99	9
4	12	22	33	99	9
4	9	12	22	33	99

I then assign the first number in my steps, which is 1,391,376, to `temp`. The nested loop doesn't trigger until `temp` is less than the size of the array to be sorted. In this example the size is 6. Thus the first nested loop doesn't trigger until `temp` is 3, meaning that the outer loop has already cycled 14 times. The first nested loop will then cycle from 4 ( $3 + 1$ ) to 6 (size of the array) times.

The `l` variable temporarily holds the value in position 4, which is 12, and `j` is made to equal 4. The second nested loop iterates while `j` is greater than `temp` and position 1, value 22, is greater than 12, moving 12 into position 1 of the array and decrementing `j` to 1 which terminates the loop as `j` is no longer greater than 3. 22 is then put into position 4, where 12 was.

The second loop iterates again with `x` now being equal to 5, `l` therefore stores 33, and `j` equals 5. The third loop iterates with 5 being greater than 3 and position 2, value 99 being greater than 33, so 99 is assigned to position 5. The `temp` variable still holds 33, `j` is decremented to 2 and the loop terminates because 2 is not greater than 3. 33 is assigned to position two.

The second loop iterates again with `x` now being equal to 6, `l` therefore stores 9, and `j` equals 6. 6 is greater than `temp` (3) but the third position in the array is occupied by 4 which is not greater than 9, so all the nested loops now terminate.

`K` now equals 16 however, and `temp` is therefore equal to 1. The first inner loop will then iterate from 2 to 6 times. `l` stores the value 33, which is now in position 2 of the array, and `j` is now equal to 2. 2 is greater than 1, and 12 is less than 33 rather than greater, so the inner loop does not trigger.

$x$  now equals 3,  $l$  equals 4, and  $j$  equals 3. 3 is greater than 1 and the value 33 is greater than 4, so 33 is moved into position three, and  $j$  now equals 2. 2 is still greater than 1, and 12 is greater than 4, so 12 is now moved into position two and  $j$  is decremented to 1, terminating the loop. The temporary value of four is then stored in the first position.

$x$  now equals 3 and so  $l$  stores the value 33 and  $j$  is made to equal 3. The inner loop doesn't iterate as 3 is greater than 1 but the value 12 is not greater than 33.  $x$  now equals 4, making  $l$  store value 22 and  $j$  is made to equal 4. The inner loop does iterate this time as 4 is greater than 1 and the value 33 is greater than 22, so 33 is moved into position 4 and  $j$  is decremented to 3.  $j$  is still greater than 1 but 33 is not greater than 33, so the loop doesn't iterate again. The temp value of 22 is then stored into position 3.

$x$  now equals 5 and so  $l$  stores the value 99, while  $j$  is made to equal 6. The inner loop doesn't iterate; while 4 is greater than 1, none of the values to the left of 99 are greater than 99.

$x$  now equals 6, and so  $l$  stores the value 9, while  $j$  is made to equal 6. This time the loop iterates, causing the values in the other positions to all be shuffled up one to make space for 9 in the second position. And that's it – the job is done.

## Summary

Although the Shell Sort is my personal favorite, often the most effective sort algorithms are known as Combo Sorts, that is a combination Sort made up of one or more of the algorithms I've discussed. For example the data could be split into groups, in the manner of the Merge Sort and then a Bubble Sort algorithm could be used to efficiently sort each group.

Sorting is a fundamental requirement for a lot of different applications. Users require data to be displayed in a variety of sort orders, be it within browses or on reports. Sorting makes finding items quicker and easier. Understanding these techniques can have other benefits also – you may find application in areas other than sorting.

I hope you've found this discussion of sorting algorithms useful. In my next article I'll branch off in a different direction and discuss tree data structures.

---

*[Alison Neal](#) has been using Clarion since 2000, whilst working for [Asset Information Systems](#) (AIS) in Auckland, New Zealand. Some years ago (at the tender age of 19) Alison graduated from the Central Institute of Technology in Wellington, New Zealand with a major in Cobol. She also has a BA in English literature and has studied Computer Science, Philosophy and Information Systems. AIS is an independent division of Asset Forestry Ltd, and has a team of five programmers developing almost exclusively in Clarion. AIS also offers web (ClarioNET) and email services for the customer who needs everything. The company has many and varied customers bridging across a wide range of industries including Telecommunications, Forestry & Agriculture,*

*Manufacturers, Military & Government, Legal & Financial, and Retail.*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2002 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.

**INVEST**  
in your own abilities

**Clarion**  
magazine

[Home](#) [COL Archives](#)

[Topics](#) > [Misc.](#) > [Marketing](#)

## **The Program's Finished. Now What? (Part 3)**

**by Braxton DeGarmo**

Published 2002-07-31

In this final article in my marketing series, I'd like to discuss the Internet as a tool for marketing. As I said in [Part Two](#), the Internet has forever changed product marketing. Yet the biggest challenge is being found by your potential customers. I'll assume here that you already have a professional-looking web site with the forms needed to capture potential customer data, as well as the features needed to allow those customers to download a demo of your software. All three factors in the previous statement are basic to your success. *Do not* shortchange the idea of a professional-looking site. A bad first impression will kill your marketing effort before it really starts.

Okay, now that you have a web site, your next question is, "How do people find it?" The first and easiest answer is to include the URL on all your correspondence, business cards, and brochures. Your web site is an extension of your business that can be used to inform current customers of updates and changes, and develop a new level of customer service. That, in turn, builds customer loyalty, and hopefully, word-of-mouth advertising (which remains the best form of advertising available).

### **Building new business**

It goes without saying that if you are a fledgling company with few customers, you need your site to build new business, not just customer loyalty. There's a caveat here, however. The Internet has proven to be a mediocre medium for building new business. We've seen one dot.com company after another fold because they expected the Internet to build their business. You still need to develop and implement a comprehensive marketing plan; one in which the Internet is but one component.

That said, let's look at ways in which you can use the Net to build business. One of the best,

cheapest, and easiest is to develop a newsletter. In a comprehensive marketing plan, this newsletter should be produced in both hard copy and electronic formats. The printed version can be snail mailed to clients without Internet access, while the electronic version should be available on your web site as well as by email. Your electronic newsletter can:

- discuss topics relevant to your industry
- introduce your staff so visitors can put a face to a name (and discover which person might best serve their specific needs)
- announce monthly specials
- establish your credibility
- develop your reputation as an industry expert
- cultivate a relationship with your subscribers
- inspire customer loyalty
- maintain regular customer contact
- and more

The focus of the newsletter, whether it goes out weekly, monthly or quarterly, is to present information on various aspects of your particular industry or business, *not* to simply promote your products or services. If all you do in your newsletter is promote your own "stuff", people will quickly add your address to the junk mail filter. While "educating" potential clients via your newsletter, you want to also subtly invite them to your web site. Most will sooner or later take you up on that invitation.

This, of course, brings up the topic of email, more specifically spam, the common term for unsolicited, unwanted email. The Net can be used to promote your newsletter and website through mass email. And whether we like it or not, this medium will be used more and more for mass marketing. A bulk mailing to millions of uninterested parties will not produce as good a result as a targeted mailing to 20,000 potentially interested parties, not to mention that spam can produce a lot of ill will. Appropriate use of the Net for mass email means responsible use within the guidelines of all state and federal laws, and mailings should be targeted just as I described targeting traditional markets in Part Two of this series.

Of course, there are those who feel *all* unsolicited email is spam, including many of you reading this article. If so, all I can say is this trend is here to stay, and you'll need your antacids by the case.

Some ISPs now require email lists to be double opt-in. This means the user signs up for the email list at your web site, and you send the user a confirmation email. The user must either click on a unique link contained in the email, or respond to that confirmation email, to be added to the lists. You will want to keep the confirmation information on record in the event that someone who has signed up for a list, and has not subsequently requested removal, complains to your ISP.

## **Search engines**

Ah yes, search engines: Yahoo, Google, Metacrawler, Alta Vista, and all of the other 250+ search engines. The problem with search engines is that the great majority of them are in business solely to develop mailing lists to sell to the mass email marketers. Of those who don't, there have been some major bankruptcies and consolidations since late 2000. Over 90% of all "search engine" searches actually take place through only three search engines, and Yahoo accounts for 60+% all by itself. So, why not just list with Yahoo? You could, but you'd be one of hundreds of millions and your site could end up on page 1053 of the search results. Not to mention that it could be as long as 3-6 months after you submit your site before it actually shows up.

The only "free" way to boost your position is to increase the number of links from other sites to yours. This often goes by the name of partnering, in which you share reciprocal links on your sites. This takes a lot of time and networking, and a year later you might find your site promoted to page 435. That's because every other knowledgeable Webmaster will be doing the same thing with all of their sites. Some webmasters have used "dummy link" pages in an attempt to fool the search engines into thinking there are tons of links to their site(s). Most of the crawlers and spiders used by the search engine companies can now detect such links and discount them.

There are, however, ways to maximize your return, if you're willing to develop an advertising budget to do so. The search engine providers have learned that there is money to be made in premium listings. These are the web sites that appear on the very first page of every search. In late 2001, Yahoo signed a contract with Overture to use their premium listing service. That service is designed in such a way that you pay a certain amount for every click-through, i.e. when someone actually clicks on your link. You, and your competition, determine that amount via a bidding process. If you have a highly specialized niche with little competition, you might get away with as little as \$.10 per click-through. A competitive position could cost \$1.55 per click, or more. There are several options for paying for this service. You can designate a specific prepaid amount and when that amount is used up, your listing disappears. (This is an ideal way to run a market test.) You can do the same with a specific monthly budget, and when that amount is used up each month, your listing disappears until the next month. And finally, you can live dangerously and leave the budget open ended. By the way, one other advantage to paying for click-throughs is that it often takes you to the head of the line for having various search engines scan your site - no more 4-6 month waits.

## **On-line advertising**

What about on-line advertising? By that I mean banner ads, e-zine and newsletter ads, and classified ads. Once the hot ticket for Internet advertising, banner ads have become a dirt-

cheap form of advertising. Likewise for e-zine, newsletter, and classified ads. The advertisers say it's the weak economy that has brought their prices down. In my opinion, the price decrease is due to the fact that people no longer pay attention to these ads, and often find them an annoyance. As for classified ads on a web site, when was the last time you responded to one?

However, this medium is not all negative. You just really need to be selective. If your product is in an industry or market for which there are only 2-3 dominant web sites, and if their prices are cheap enough, it might be worth a test market. Just make sure you have accurate stats on the number of *unique* visitors who visit the site. Don't get blown away by stats showing huge numbers of daily hits.

Of the three formats mentioned, e-zine/newsletter ads is the best. The readers of such e-zines and newsletters are pre-targeted for you, much like readers of a specific print magazine. Better yet, if you have a successful newsletter of your own, mutual or shared advertising might be arranged to allow both newsletters to extend their reach.

## **Affiliate programs**

If you are serious about doing business on the Internet, and have a budget that allows for more than a newsletter, it pays to have an affiliate program because such programs provide a virtual army of salespersons sending traffic to your site day and night. No matter what you are selling, there are other netrepreneurs who will eagerly promote it for you on their web sites and in their e-zines in exchange for a "piece of the action." An affiliate program works when others place a link on their web sites to your sales site, allowing them to earn commissions without any of the usual hassles of operating a "brick and mortar" business: shipping, help desks, inventory control or charge-backs.

Setting up your own affiliate program is simple. It does, however, require some good software to track links and commissions. To learn more about it, [follow this link](#) to 50+ pages of articles on the who, what, why and where of starting an affiliate/associate program. And if you visit the site, take a good look at it from the perspective of Internet marketing, since it employs many of the things discussed in this article. The purpose of the site is to promote the company's own affiliate tracking software, but they do so by providing a site full of information on their "industry" with mostly subtle references to their product. (And, no, I'm not an affiliate and make no commission from referring you there.)

## **A final note ...**

One final method of promoting your site is to write articles like this one. By providing material for e-zines and newsletters, you can establish yourself as an "expert" in the field while attracting visitors to your site, which is typically listed in your brief bio at the end of the

article.

---

*[Braxton DeGarmo, MD](#) has been programming with Clarion since 1989. As a practicing physician, he needed the most efficient development tool possible to continue his medical programming, and Clarion was the perfect prescription. In the past two years, he has broadened his programming to several vertical markets, including sportfishing and fiction writing.*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2002 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.



For marketing your Applications  
and Developer Accessories or to  
purchase other 3rd Party Tools ...

Developer  
**PLUS**<sup>tm</sup>

[Home](#) [COL Archives](#)

[Topics](#) > [Design & Development](#) > [Clarion IDE](#)

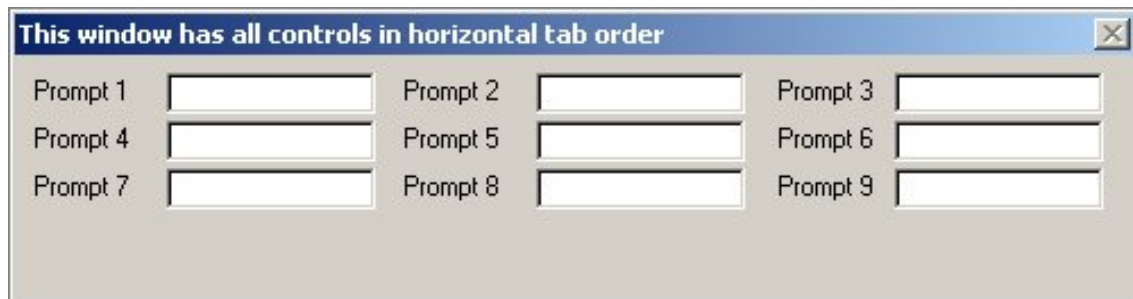
## Shrink-Wrapped Controls

by **Jeff Slarve**

Published 2002-07-31

There's a really cool feature in the C5.5 window formatter that you might not know about. It's called "shrink wrap," and I will show you a great time-saving way to use it.

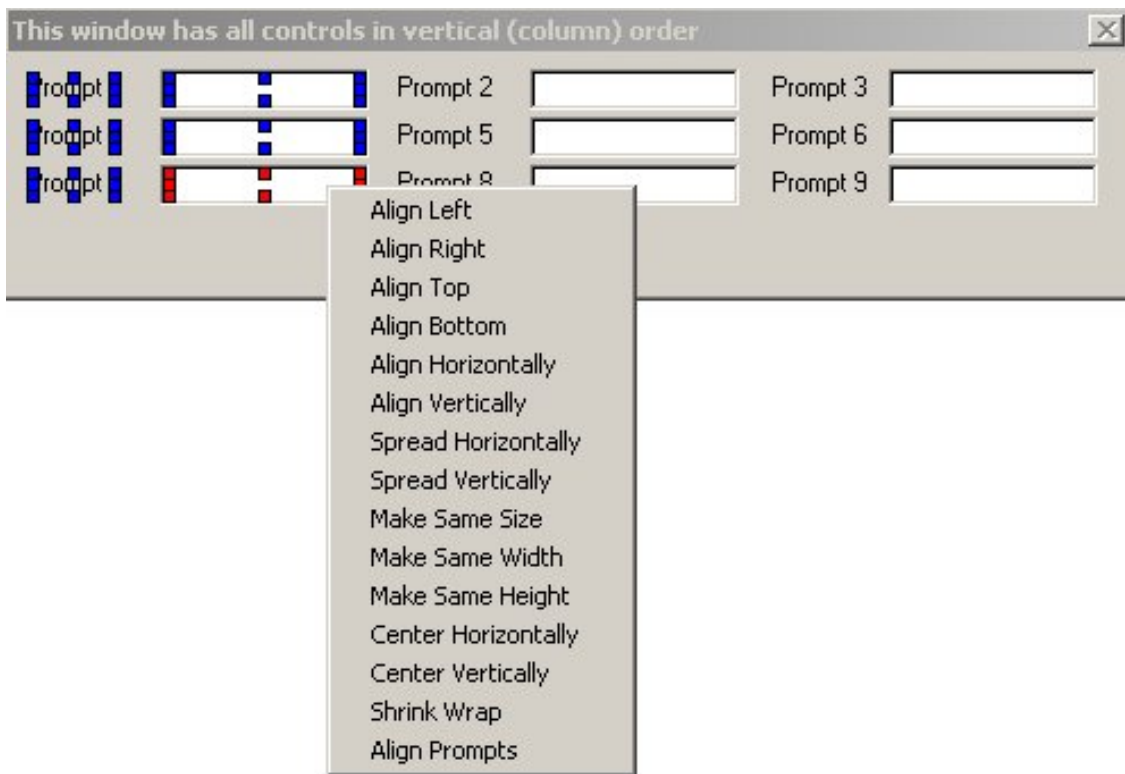
Some of the more common questions that I've seen in the newsgroups involve setting the tab order of controls on a window. Although the IDE does a fairly good job of guessing where you want a control's position to be, it can't always know the intentions of the software developer. That means you have to change the tab order manually, either by pressing F12, or by going into the window source. Take for example, the window in Figure 1.



**Figure 1. Controls on a window**

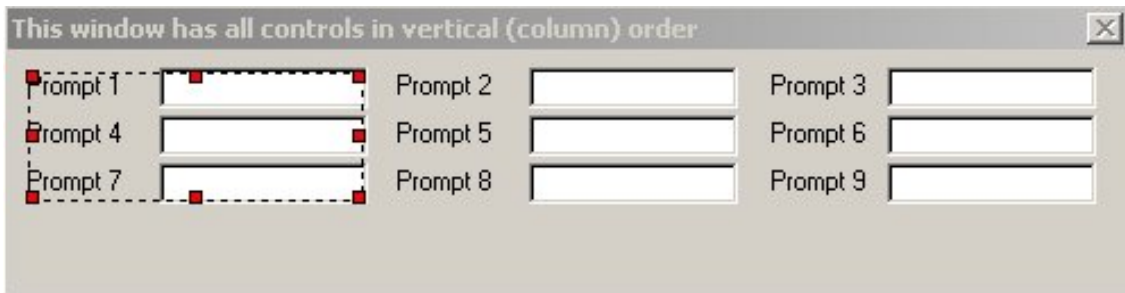
In this window, the tab order of the controls goes horizontally, one row at a time. What a pain it is to re-order these controls if you want the tab order to go vertically, one column at a time. Whether you make the changes by pressing F12 or by editing the source, it's very time consuming and tedious work.

Well, never fear. It's time to shrinkwrap those columns. It's really simple. All you do is hold your CTRL key down and lasso the first column of controls, as in Figure 2.



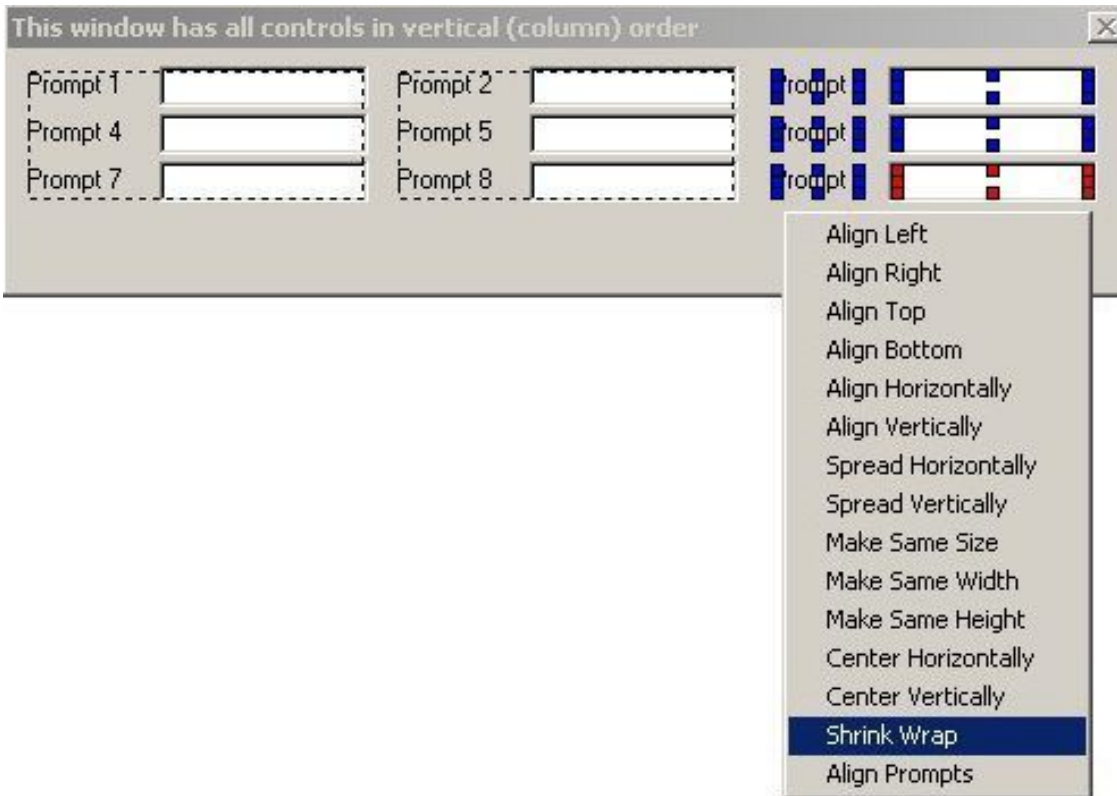
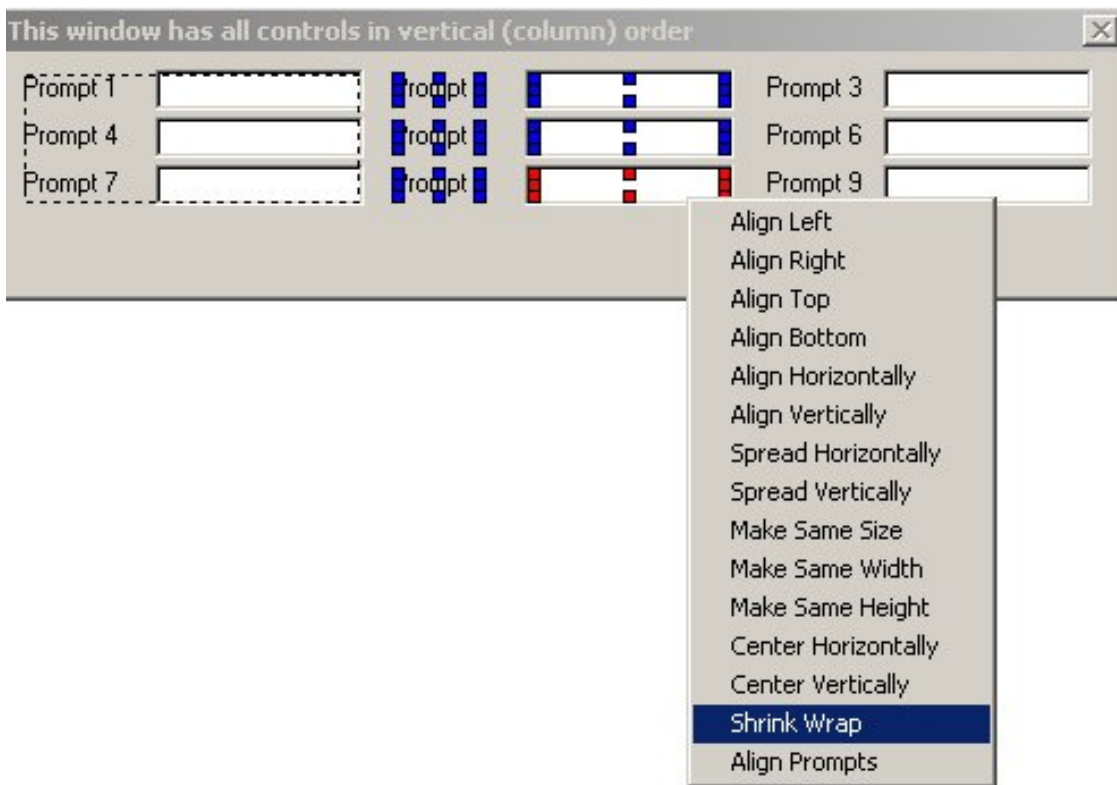
**Figure 2. Shrinkwrapping the selected controls**

Right-click on one of the lassoed controls and select Shrink Wrap from the popup menu. This creates a GROUP structure around the selected controls, and also sets the tab order within the new GROUP, as shown in Figure 3.



**Figure 3. The controls with a shrinkwrapped GROUP**

Now just repeat the same process for the second and third columns:



It's that easy.

Now, if you want to get rid of those GROUP controls, it's a pretty simple task to go into the

window text and delete the GROUP lines and their corresponding ENDS.

For complicated windows, this trick has saved me lots of time.

[Download the source \(example only, not running code\)](#)

---

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

## Reader Comments

[Add a comment](#)

Copyright © 1999-2002 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.

For marketing your Applications  
and Developer Accessories or to  
purchase other 3rd Party Tools . . .

Developer  
**PLUS**<sup>tm</sup>

[Home](#) [COL Archives](#)

[Topics](#) > [Tips/Techniques](#) > [Clarion Language](#)

## Reports: OOP, ABC and Ignoring Templates (Part 1)

by **Steven Parker**

Published 2002-07-31

Preparing my ETC presentation, "Report: Paying the Price of Flexibility," I had an epiphany.

While not as painful as most such events, I was, nevertheless, thankful that it was short-lived. I recovered quickly, though some question remains about how thoroughly.

The revelation? I don't *do* OOP. I *use* OOP, as provided by the ABC class libraries, but I do not *do* OOP. I have not created a single class, much less a class library. I do have an occasional attack of "I really should learn this stuff." But, that would require doing, and I haven't been *that* tempted. Moreover, that means that I use templates, the ABC template chain, particularly, strangely.

I just haven't yet felt the need. I have yet to find anything I need to do or want to do that I cannot do quite easily with my tried and true techniques of incorporating frequently used procedures in DLLs and INCLUDE-ing blocks of frequently called code.

In fact, I have recently become seriously enamored of the optional Section parameter of the INCLUDE statement. This INCLUDE variant allows me to selectively include parts of a single INCLUDE file. Most useful, this, for creating multiple embeds out of the same core code but without inhibiting complete customization. It provides something very much like embeds in an INCLUDE.

For example, I have an application whose behavior is entirely defined by ALERT keys. The main procedure is little more than:

```
Case KeyCode()  
Of ThisKey  
!ThisKey stuff
```

```
Of ThatKey
  !ThatKey stuff
!Etc.
```

Actually, this *is* the main processing loop. The only action in this app *not* called from this CASE structure is "help." Even "exit" is controlled by this structure.

It turns out that several of the keys call behaviors that are remarkably similar, virtually identical in fact.

A case in point would be a set of tendering keys in a POS application. Paying by cash, by check (or, even, cheque) or by credit card differ only in the teeny, tiny details.

Needless to say, the handler code is remarkably similar too. The expected behaviors differ only in some initial error checking, specific to what the user is asking my program to do and, later in the process, in priming different local variables.

As 90 or more lines of code are involved and 15 or 18 keys may be in this "family," my lazy programmer alarm went positively berserk. I suppose I could have created a class, incorporated all the needed behaviors, made runtime "decisions" based on parameters and called the class method(s) at the appropriate time.

Using `INCLUDE(filename[,section])` I ended up with:

```
Of ThisKey
  !error checks
  Include('includeFile','FIRST')
  !another unique behavior, if required
  Include('includeFile','SECOND')
  !load local variables
  Include('includeFile','THIRD')
Of ThatKey
  !error checks that are unique to this action
  Include('includeFile','FIRST')
  !another unique behavior, if required
  Include('includeFile','FOURTH')
```

And in the `includeFile`:

```
SECTION('FIRST')
  ! Load up some standard variables here

SECTION('SECOND')
  ! Some source code specific to ThisKey processing

SECTION('THIRD')
  ! Load some local variables

SECTION('FOURTH')
```

! Some source code specific to ThatKey processing

(Isn't my creative naming convention impressive?)

My embedded code is reduced to under 35 lines and *all* the common code is maintainable in one place. And I don't even have to cut and paste 15 - 18 times.

Not a class, method or property in sight. And I shudder to think how large (and logically convoluted) a class incorporating what I need would have been. (I remain convinced that OOP, generally, and the lower-level kind of implementation represented by the ABC classes, specifically, is the ultimate in pasta code. Nature of the beast, to be sure, but I just can't wrap my brain around it – running all over a source file, or more than one, to track what's going on, is just not my idea of fun.)

I am not, by any stretch of the imagination, saying that I'll never need to write a class. I am saying that I haven't felt the need to just yet. In other words, I am, despite using ABC exclusively, an unrepentant procedural programmer. I am hopelessly linear.

However, after considering the matter, not only wasn't I very surprised by this sudden enlightenment (I suspect many Clarioneers develop in a similar manner: you use OOP but many of you don't "do" it), I found it dovetailed well with the way I have always used Clarion.

And it is the way I think about creating procedures from templates that I want to talk about.

The Clarion template system has always provided a very substantial portion of my procedures (note, only "a substantial portion"). And, over the years, more and more functionality has been included in the base template procedures (okay, to be entirely accurate, into the ABC classes). In fact, the ABC templates provide a dizzying array of embeds (the vast majority of which, frankly, have no conceivable use – still, it's nice to know they're there) and a plethora (yes, a real, honest to goodness, full fledged plethora) of methods and properties so extensive that, even after the years ABC has been available, Clarioneers are still discovering things those classes can do for them.

My approach to template-based procedures, I long ago realized, has been to take what the template gives me (or, at least, the parts that work as I expect/need/want them to), ignore the rest to the degree possible and supply the remaining behaviors myself. I happily use Procedure Wizards to create new procedures, spend some time (often quite a lot) on aesthetics and promptly go into the embed tree and place a comment in the embeds I know I'm most likely to use. That way, I can contract the embed tree and still have immediate access to my favorite embeds (TakeRecord, TakeNoRecord, INIT and the like). Thereafter, I look at the window formatter only for reference or to add something the wizard "forgot."



Sometimes I even remember to remove the comments from the embeds I did not end up using.

If this is my attitude toward template procedures, it shouldn't surprise anyone who has followed my written musings over the years, or my newsgroup participation, that I haven't much patience with questions like:

- Why doesn't the template do this for me?
- The template should offer an option for ....
- The template should allow me to override ....
- The IDE should be smart enough to ....
- It's a bug that <whatever> doesn't happen.
- It's a bug that <same "whatever" as above> *does* happen.

Bugs? If I can code around it, I can't be bothered waiting for a fix; time, tide and my boss wait for no man. And there haven't been many that couldn't be coded around.

## **What the Templates Provide: Essential Features**

So what, exactly, is it that I expect the templates to provide? What is it that I do not ignore?

At the core, the templates provide me with:

- a data section
- a formatter (window and/or report)
- a View (where appropriate)

and

- a processing loop with lots of places for inserting my own code.

The Report Template adds a PRINT statement in the processing loop. I don't suppose that that is very surprising.

Many of the templates also throw in file handling. But that's another story entirely.

Remarkably, that's only four items I consider truly important. Everything else in a template hangs off these four items but, with these four alone, I can do just about anything I need to using only the Clarion language and ABC methods (ok, a third party template here or there ...).

**Data Section:** one of the defining characteristics of Clarion has been the Data Section in each procedure. The Data Section is more than a place where local variables are declared; Clarion has always treated screens as data. The Data Section contains any screen layouts, whether provided by a template or not. And, because it is data, a screen can be manipulated by



assignments, just like any other data, after memory has been allocated (i.e., it has been OPENed).

A report is just a variant on a screen. So it, too, is a data structure. (It has the added issue of requiring a `SetTarget` before an assignment but that, too, is another story.)

**Window/Report Formatter:** Clarion's first claim to fame was the fact that it integrated a screen/report painter into its text editor. This allowed drawing screens and reports without having to remember high order ASCII characters or brush characteristics and count spaces and lines ("Hollerith counts" anyone?). It was there in 1.0. It's still there in 5.5.

The formatters allow me to create my windows and reports visually and they generate the code for me. It's one of the things we all love about the Clarion IDE.

**A View:** In procedures where a primary file is required, usually for sequential reading (browses, reports and processes, for example, but not forms, frames, source and window procedures), the templates create a View. It Projects, it Joins, I view.

The important thing about the procedure's View is not that it is a virtual file, though it is that, not that it can contain fields from multiple files (based on relations), though it can do that, and not that, once filled, is very fast.

The important thing about the View is the properties that can be associated with it. The key properties are the `FILTER` and `ORDER` attributes.

By themselves, these attributes would be nice but not especially useful. The Language Reference Manual states that these attributes modify the `VIEW` declaration itself. To be flexible enough to be truly useful, these attributes would have to accept variables (and these variables would have to be primed before the view was OPENed). This would allow dynamic filtering and ordering.

They don't support variables (that's Ok because having to prime them before the view opens would seriously limit my creative options). But, they don't have to. The View Engine has methods impacting the filter attribute and the order clause.

This is where I get the ability to set and re-set browse and process filters and sort orders at runtime. And that is why I consider the View engine the single most important feature provided by Clarion and the view-related methods the most important thing supported by the ABC templates.

**A Processing Loop:** Clarion has always been "event driven." That is, it has always processed screens and windows in a loop, `ACCEPT`. This allows forms to check and act when a field is

completed and when one is selected. It allows a batch process to read records sequentially and reports to print them.

In a batch, the record is read in this loop. A Report adds a PRINT after the read. A form checks CASE Event ( ), CASE Selected ( ) and CASE Accepted ( ) to provide field level processing.

This is not rocket science. Even those, like myself, not trained as programmers can handle loop processing. Most templates add lots of bells and whistles to make this more convenient for us (like the three CASE structures in a form) but the bottom line is "It's the loop, dummy."

## Summary

I've made a fairly outrageous statement, that except for the data section, the formatter, the view and the ACCEPT loop, it is entirely possible to ignore the balance of a template generated procedure.

Now let me go a step further and say that it is often wise to do so.

In the next installment, I'll justify these statements by walking through the Report Template and demonstrating the liberating effect of ignoring the template.

---

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

## Reader Comments

[Add a comment](#)

**What an awesome idea! It will change the way I do...**