

INVEST
in your own abilities**Clarion**
magazine[Home](#) [COL Archives](#)

ETC III: SQL Replication - Shawn Mason

Shawn Mason gave an excellent presentation on SQL replication (and SQL in general) at ETC III. James Cooke reports.

Posted Monday, June 10, 2002

Positioning List Box Totals

Sometimes you'll want to place total fields below list boxes, and in some cases those totals need to be positioned exactly below a column. This is easy if the user doesn't resize any columns, but what if you do allow resizing? Brice Schagane shows how to align the total fields with the columns at runtime.

Posted Tuesday, June 11, 2002

Interfacing With An Interface

Interfaces are becoming increasingly important in the evolution of ABC. In this article, Phil Will shows how to implement the WindowComponent interface to make the currently selected control more visible to the user.

Posted Wednesday, June 12, 2002

ETC III: Various Photos

Thought we were done with ETC photos? Not quite - here's one last batch, with contributions from Bruce Johnson, Brice Schagane, Brad Noble, and Nik Johnson.

Posted Thursday, June 13, 2002

Data Structures and Algorithms Part III - Maintaining Order

Alison Neal continues her series on basic data structures and algorithms with a discussion of list sorting, beginning with the Bubble Sort.

Posted Wednesday, June 19, 2002

The Checkbox Class Performance Upgrade

Last year David Harms wrote an article about using browse checkboxes to manage many-to-many relationships. Faced with using this code on a remote database via an Internet connection, he found the performance unacceptably slow. This redesigned class dramatically reduces network traffic.

Posted Thursday, June 20, 2002

News

[SetupBuilder SetupScript FAQ](#)

[XPMenu Lookalike Developments
And Holiday Schedule](#)

[SysTree 1.1 Released](#)

[CapeSoft MessageBox 1.6d](#)

[Secwin 3.16d Beta 6](#)

[TickerTape 1.61](#)

[Insight Graphing 1.0 beta 7g](#)

[HyperActive 1.7 Update](#)

[Special Agent 1.41](#)

[File Explorer 2.4c](#)

[File Manager 3 Limited Release](#)

[Programming in Clarion's ABC -
Book Release](#)

[xAccess Manager Lite v1.2](#)

[Imaging Templates Version 1.14
Released](#)

[INN Bio For 20-June-2002](#)

[EmailReport 2.5 Released](#)

[xWhatsNew v1.1 Released](#)

[Minor Update To FSData Class](#)

[EmailData 1.3 Released](#)

[Clarion Third Party Profile Exchange
Updated](#)

[Jack Toering On Network Operating
Systems](#)

[Scripting Language For C55](#)

[Latest D2MySQL Templates And
Example](#)

[Coollook Template Demo](#)

The RanDum Phrase Generator

Here's a neat little program by Lee White that generates random phrases, suitable for security applications such as automatically generated passwords.

Posted Friday, June 21, 2002

Data Structures and Algorithms Part IV - More Maintaining Order

Alison Neal continues her series on basic data structures and algorithms with a discussion of the Cocktail Shaker, Insertion, and Counting sorts.

Posted Wednesday, June 26, 2002

Emailing PDF Files From A Report Previewer

In recent months, a number of third-party products have come into the marketplace that allow the user to save a report in a standardized format, and then email that report to someone. But according to George Lehman, there's still something all these products are missing.

Posted Friday, June 28, 2002

Clarion And The Fatal MyODBC/MySQL Flaw

David Harms has been a proponent of the MySQL open source database for a few years now, but recent performance problems have him looking for an alternative to the MyODBC driver and, consequently, MySQL.

Posted Friday, June 28, 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.

[chSTD Library v2.6 Released](#)

[EasyReport v1.01 Released](#)

[Arnor & Sue In Iceland](#)

[Freeware Reports-To-WMF Template Updated](#)

[Clarion/ASP Support And Training Newsgroups Online](#)

[XPMenu Extra Demo](#)

[Freeware: Create XLS Files In Clarion](#)

[Clarion/ASP Video](#)

[Bruce Johnson's ABC Notes Now Available](#)

[Free XML Wrapper](#)

[SterlingData Half Price VCRFlash Until June 14](#)

[etc-III Lodestar Software Discounts](#)

[Free MP3 Player](#)

[Gitano Software June Drawing](#)

[RichReport Gold](#)

[Coollook Template Release v2](#)

[Clarion/ASP Online Training Course](#)

[Free Plugware COM Classes Available Now](#)

[EasyExcel 1.031 Released](#)

[XPMenu At ClarionShop](#)

[June Simsoft Template Special](#)

[PDF-Tools & Clarionet Reports 2 PDF Demo](#)

[New "Look Good" Package](#)

[Interactive ASP Tutorial](#)

[Clarion 5.508 \(h\) And ClarioNET 1.2 Available](#)

[Clarion/ASP Released](#)

[SealSoft's xFunction Library v1.5](#)

[DCT2SQL Templates Update](#)

[XPMenu Online Purchase](#)

[Search the news archive](#)

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](#) > [DevCons](#) > [ETC III](#)

ETC III: SQL Replication - Shawn Mason

by James Cooke

Published 2002-06-10

Over the last 10 years or so Clarion has evolved from a small-scale database development environment into a powerful, scalable development environment that rivals the high-end systems. Clarion 5.5 Enterprise Edition ships with 15 database drivers, enabling Clarion programmers to interface with almost any database system. This has very effectively shielded the programmers from the majority of nuances that proliferate in these backend environments, thus enabling them to get on with the business of programming.

The problem that this has brought about is that many Clarion programmers today have next to no understanding of interfacing with external database environments. Shawn Mason's excellent discourse covered not only Microsoft's powerful Database Replication technology, but also provided the audience with an outstanding introduction to what SQL is, why we need it, when we need it, and all the juicy bits and pieces as to why we want it! This article covers the main points of Shawn's presentation.

The first thing that seasoned Clarion programmers ask, before moving to SQL, is "why bother?" Because the truth is this: Within the scope of the small to medium sized project, ISAM files (Topspeed/Clarion) give us what we need, reliably and quickly. But the scope of the project will be limited to just that: small to medium sized applications. Move to bigger databases, more users, and higher volumes, and ISAM starts having problems.

SQL provides scalability, in that for single user, low traffic, low volume databases it is fine, and for multi user, high volume and high traffic databases it is also fine. The SQL language itself is also very powerful - for example, it supports aggregate queries, server side summary reports and stored procedures (which are data manipulation programs that run on the server instead of the client). All of these factors may reduce network traffic by several orders of magnitude.

The primary issue that stops Clarion programmers from using SQL across the board is cost, which is usually on a per-seat basis, whereas Clarion's ISAM database support is free.

Microsoft supplies a comprehensive suite of management tools that provide the DBA (database administrator) with enough oomph to do more or less what she pleases. These tools include Enterprise Manager (similar to the Clarion dictionary ide), which allows the user to define tables and fields, browse and modify data files, transfer data/tables/databases between servers, DTS (export data to external data sources), query analyzer (allows the user to browse data tables using custom SQL), and the replication manager.

The replication manager provides the programmer with the ability to control the data on a distributed system. Data can be split across different locations, to eliminate network traffic and reduce load on a single server. It also allows one to move data off a central server and onto other servers, or notebook computers for example, to provide for high availability and/or decentralization of traffic. A typical example would be a point of sale system in a busy retail outlet: All PCs run basically standalone, and every three hours there is a synchronization of data where the data on the main server (prices etc) is pumped down to the POS systems, and the POS systems sales data is pumped back to the server for sales analysis/reordering etc.

The replication engine manages all the complex issues of managing primary key uniqueness, with these keys cascading down to foreign keys etc. This is something no sane programmer would want to get into - as things rapidly mushroom into a tangle of referential spaghetti! Shawn gave us a step-by-step guide on what type of replication to setup (snapshot/transactional/merge), and provided various real-life scenarios, and how the replication manager would resolve each.

Robert Zauhere's demo of C5.6's integrated SQL support shows that Clarion's orientation might be shifting toward supporting a true Client/Server paradigm - so this presentation was timely indeed. Shawn's preparation was thorough, and his presentation was just the sort of springboard that many Clarion programmers need for moving into the main stream!

[James Cooke](#) works for Farm Credit Bank of Texas, and is currently developing several systems on an AS/400. He has been developing Clarion applications since 1992, and lives in Austin, Texas with his wife and two young children. For relaxation, he enjoys medium format photography, balanced with a liberal dose of sun next to the pool!

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
PLUStm

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

[Topics](#) > [Browses](#) > [Browses, Using](#)

Positioning List Box Totals

by Brice Schagane

Published 2002-06-11

Sometimes, I place total fields below list boxes, and in some cases those totals need to be positioned exactly below a column. This is easy if the user doesn't resize any columns, but as I do allow resizing I needed a way to align the total fields with the columns at runtime.

For example, I have an application used to keep track of transactional information. Each transaction contains an amount field. In my list box, I split the amount field into two columns, one for Credits and one for Debits. I also have two total fields on my Browse window, one for Total Credits and one for Total Debits. I would like each of the total fields to be displayed directly under their corresponding columns, as illustrated in Figure 1. Resizing any column within the list box will cause the Credits and Debits columns to be repositioned. If this happens, I need to reposition my total fields as well. In this article, I will explain how this can be done.

Number	Date	Payee	Debit	Credit	Balance
	1/01/01	Opening Balance		100.00	100.00
101	1/02/01	Steve Parker	10.00		90.00
102	1/15/01	The Rent	15.00		75.00
103	1/15/01	Electric Co.	10.00		65.00
110	1/15/01	Mortgage Co	50.00		15.00
	1/20/01	Deposit		25.00	40.00
	1/30/01	Deposit		50.00	90.00
TOTALS:			85.00	175.00	

Figure 1. Display of Total Debits and Total Credits

Working with positions

In order to set the position for my total fields, I need to know the position of their related list box columns. In Clarion, field positions and sizes, within a list box, are not available as a simple property. However, the fields can be referenced using field equate labels. Each field within a list box has a unique field number. I need to know the field numbers before I can reference any of the fields. Using Clarion's list box formatter, I can obtain the numbers for each field. Field numbers can be assigned automatically by Clarion or I can assign them myself. Field numbers assigned by Clarion begin with one and continue in ascending order, based on their placement within the list box. I can get each field number by simply counting the fields in the order of their placement or by assigning them myself. If I were to rearrange the fields within my list, then the field numbers could change and I would need to modify my code. For this reason, it would be best to permanently assign the field numbers.

Once I know the field number, I can use it to help determine the fields' position and size. Since, these properties aren't available to me directly, I will use an edit-in-place trick to find them. EIP will allow me to assign a window control to any list box field for the purpose of obtaining input from the user. I can easily obtain positional properties from a window control. This is illustrated by the following code:

```
!Create Region
Control_ID# = Create(0,Create:Region)
!Assign Region to List Box field
?Browse:1{PROP>Edit,5} = Control_ID#
```



```

!Set Position
?TotalDebits{PROP:XPos} = Control_ID#{PROP:XPos} |
  + ?Browse:1{PROP:XPos}
!Set width
?TotalDebits{PROP:Width} = Control_ID#{PROP:Width}
!Unassign the Region
?Browse:1{PROP>Edit,5} = 0
DESTROY(Control_ID#)

```

First I create a REGION control. Then I assign the region to my field in the list box as if it were an edit-in-place field. By doing this, Clarion automatically moves and resizes the region control over the field indicated, within the currently selected row of the list box. The X position of my region control is now the same as the X position of my list box field. I'm not interested in obtaining the Y position of the field, so it's not important which row is selected.

Next, I assign the new position for my TOTAL field. Since the region control is assigned to a field within the list box, its position is relative to the upper left corner of the list box. I must take this into consideration when setting the position of my TOTAL field. For this reason, I set the X position of my TOTAL field equal to the X position of the region control plus the X position of the list box. I then set the width of my TOTAL field equal to the width of my region control, which is the same as the width of the list box field.

Let's say I wanted to vertically align a graphical pointer with the selected row in the list box. To do this, I would need to obtain the Y position of the region control. Also, I would have to account for the Y position of the list box. For example, setting the region control's Y position to zero (0) would place it directly over the column header. If I only have one line of data per row in the list box, then obtaining the height of the region control would also give me the height of the row. If I were using a multi-line list box, then I would have to consider the height multiple fields in order to obtain the height of the row.

Finally, I unassign the region control from the list box field and then destroy the region control, since I only needed it to get the column location and width.

Embedding the code

Instead of duplicating my code in multiple locations, I simply wrote a single routine named MoveTotals.

There are two events which must cause my code to execute. The first is when the window is initially opened, and the second is when any column within the list box gets resized.

In the ThisWindow.Ask method, before the parent call, I embed the following:

```
Do MoveTotals
```

In the `ThisWindow.TakeFieldEvent` method for `EVENT:ColumnResize`, after the parent call, I embed the following:

```
Do MoveTotals
```

If I did not allow my users to resize any columns within the list box, I would not need any code to handle resizing.

Cleaning up

If I allow any columns to be resized, then it is possible that the columns, used for totaling, could extend beyond the view of the list box. I must include code to handle this condition, when it exists. This can be accomplished using the following code:

```
IF ((?TotalDebits{PROP:XPos} + ?TotalDebits{PROP:Width}) > |
    (?Browse:1{PROP:XPos} + ?Browse:1{PROP:Width})) Then
    HIDE(?TotalDebits)
ELSE
    UNHIDE(?TotalDebits)
END!_IF
```

This code compares the right boundary of the TOTAL field against the right boundary of the list box. If the right boundary of the TOTAL field extends beyond the right boundary of the list box, then I will hide the TOTAL field from the display.

For some unknown reason, my debits total is sometimes duplicated overtop of my credits total. To prevent this condition, I simply included a `ThisWindow.Reset(1)` at the end of my `MoveTotals` routine.

I also have a `Totals:` label for my TOTAL fields. I included code to adjust its position with my total fields. The final code for my `MoveTotals` routine looks like this:

```
MoveTotals          ROUTINE
!Move Debits
!Create Region
Control_ID# = Create(0,Create:Region)
!Assign Region to List Box field
?Browse:1{PROP>Edit,5} = Control_ID#
!Set Position
?TotalDebits{PROP:XPos} = Control_ID#{PROP:XPos} |
    + ?Browse:1{PROP:XPos}
!Set width
?TotalDebits{PROP:Width} = Control_ID#{PROP:Width}
IF ((?TotalDebits{PROP:XPos} + ?TotalDebits{PROP:Width}) > |
    (?Browse:1{PROP:XPos} + ?Browse:1{PROP:Width}))
    HIDE(?TotalDebits)
ELSE
```

```

    UNHIDE(?TotalDebits)
END!_IF
!Unassign the Region
?Browse:1{PROP:Edit,5} = 0
!Destroy Region
DESTROY(Control_ID#)
!Move Credits
!Create Region
Control_ID# = Create(0,Create:Region)
!Assign Region to List Box field
?Browse:1{PROP:Edit,6} = Control_ID#
!Set Position
?TotalCredits{PROP:XPos} = Control_ID#{PROP:XPos} |
    + ?Browse:1{PROP:XPos}
!Set width
?TotalCredits{PROP:Width} = Control_ID#{PROP:Width}
IF ((?TotalCredits{PROP:XPos} + ?TotalCredits{PROP:Width}) > |
    (?Browse:1{PROP:XPos} + ?Browse:1{PROP:Width}))
    HIDE(?TotalCredits)
ELSE
    UNHIDE(?TotalCredits)
END!_IF
!Unassign the Region
?Browse:1{PROP:Edit,6} = 0
!Destroy Region
DESTROY(Control_ID#)
!Move TOTALS label
?STR:Totals{PROP:XPos} = ?TotalDebits{PROP:XPos} |
    - ?STR:Totals{PROP:Width} - 4
IF ((?STR:Totals{PROP:XPos} + ?STR:Totals{PROP:Width}) > |
    (?Browse:1{PROP:XPos} + ?Browse:1{PROP:Width}))
    HIDE(?STR:Totals)
ELSE
    UNHIDE(?STR:Totals)
END!_IF
!Reset the Display
ThisWindow.Reset(1)

```

Summary

This article should give you a good handle on aligning controls with columns in your list box. It also provides basic know how, in case you are interested in aligning controls with rows in your list box.

Contributions

Special thanks to Steve Parker for allowing me to use and modify the code from his article on [Running Totals](#).

[Download the source \(Clarion 5.5 only\)](#)

Brice Schagane works for the Kentucky Transportation Cabinet. He also runs a small computer company by the name of Ghost Solutions, Inc. Brice has been using Clarion since 1997.

Reader Comments

[Add a comment](#)

**Brice: Good trick. Now I need to go change one of my...
I noticed in the example app that if I resize a column,...
This is a pretty cool idea. It's also a good example of...**

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**tm

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

[Topics](#) > [Forms](#) > [Forms, validation](#)

Interfacing With An Interface

by Phil Will

Published 2002-06-12

C55 expanded Clarion's implementation of Object Oriented Programming with the introduction of the INTERFACE structure. As was made clear at the ETC III conference in May of 2002, interfaces will become increasingly important to expanding the functionality of Clarion applications. This article highlights an example of an interface in the ABC library, using it to illustrate some interface attributes.

During his ETC keynote address, Bob Zaunere talked about some new features of Clarion 5.6, one of which concerned making it clearer to the users what field on a window had focus. The interface example developed here takes this idea and applies it to C55, using the ABC library WindowComponent interface as a way to hook into functionality built into the WindowManager class.

Functional specification

Functionally, I decided I would like to write a control selection class that made it clear to the user what control has focus, and whether the field is required. To accomplish this, I came up with the following specifics:

- All required entry, spin, drop combo, and text fields on a window would be colored red.
- All non-required entry, spin, drop combo, and text fields would be colored yellow when they have focus.
- All required fields would be colored something between yellow and red when they have focus.
- All controls, including lists and buttons, would be surrounded by a box when they have focus. The box would be gray for those that are not required, and red for those that are.

Knowing that the ABC library has declared an interface called “WindowComponent,” and knowing a bit about its methods, I was pretty sure I could use this to limit the number of lines in each procedure to two. I also wanted a simple global template that would add the code to every procedure with a window in an application, so I wouldn’t have to embed any code at all.

Interface defined

For those who may not be familiar with an interface here is the brief and very simple Clarion definition: *An interface is a collection of methods implemented by a class.*

Some of the attributes of an interface include:

- The structure is declared as an `INTERFACE`
- It can not have any properties.
- Its methods are implicitly `VIRTUAL`
- Interfaces themselves implicitly have a `TYPE` attribute.
- Unlike the methods of a `CLASS`, the methods prototyped in the `INTERFACE` need not be coded.
- Interfaces may be derived from other interfaces, but the parent interface’s methods cannot be overridden.
- Among other things, interfaces may become parameters to methods, elements in a queue, and properties in classes.
- To implement an interface, you add an `IMPLEMENTS(InterfaceName)` attribute to a class definition.
- Interfaces may have more than one implementation, i.e. they may be implemented in more than one class.
- In a class implementing an interface, all the interface methods must be coded (even if only a stub).

Interfaced methods are declared using dot syntax, specifying the name of the class, the interface, and the method:

```
classname.interfaceName.methodname PROCEDURE
CODE
```

WindowComponent Interface

The `WindowComponent` that I will use as a hook into the `WindowManager` code is an interface declared in `ABWindow.inc`, as follows:

```
WindowComponent INTERFACE
Kill PROCEDURE
Reset PROCEDURE(BYTE Force)
ResetRequired PROCEDURE, BYTE
SetAlerts PROCEDURE
```

```

TakeEvent      PROCEDURE , BYTE
Update         PROCEDURE
UpdateWindow   PROCEDURE
END

```

The `WindowManager` itself contains a queue of interfaces as one of its properties, and a method for adding interfaces to the queue, declared as follows;

```

ComponentList  QUEUE , TYPE
WC             &WindowComponent
END

CL            &ComponentList , PRIVATE
AddItem       PROCEDURE ( WindowComponent WC )

```

If you examine the code within the `WindowManager` class (`Abwindow.clw`), you will find for each of the methods where `WindowComponent` and `WindowManager` method names are the same, the `WindowManager` loops through the component queue and calls the appropriate component method. For example, in the `WindowManager.TakeEvent` method, you will find the following code:

```

LOOP I = 1 TO RECORDS ( SELF.CL )
  GET ( SELF.CL , I )
  RVal = SELF.CL.WC.TakeEvent ( )
  IF RVal THEN RETURN RVal.
END

```

The effect of this code (which is called on an event such as a mouse click) is to loop through all the window components that have been passed to the window manager with the `AddItem` method, calling its code. The called method must be from a class that implements the `WindowComponent` interface. One way to think about this is that you can have the window manager call your code without adding code in more than one embed point.

The `WindowComponent` interface gives the `WindowManager` a simple yet powerful tool to do a complicated job. It intercepts all events via its `ACCEPT` statement and uses the interface mechanism to pass events on to multiple browses, filedrops, and things like my field select class, even though it doesn't know about these classes at compile time. With the `WindowComponent` it has a known signature or subset of methods which it can reliably call. In Clarion 5, there is no comparable hook; the `WindowManager` had predefined queues of `FileDrop` and `BrowseLists` with the `PRIVATE` attribute, and no easy way of adding anything new. These queues disappeared in C55, replaced by the queue of `WindowComponent` interfaces.

Implementing an interface

To implement an interface, you must add the interface as a parameter to the `IMPLEMENTS`

attribute of a class. My control selection class has the following declaration:

```

PDSelectCT      CLASS,TYPE, IMPLEMENTS(WindowComponent), ←
  MODULE('PDSelect.clw'),LINK('PDSelect.clw',1)
LastField      LONG,PROTECTED      ! The last field with focus
LastColor      LONG,PROTECTED      ! The color of the last field
                                   ! before it was changed.
CurrentField    LONG,PROTECTED      ! The control that currently has focus.
BoxFEQ         LONG,PROTECTED      ! Field equate for the box around the
                                   ! control with focus.
TakeFieldEvent PROCEDURE,PROTECTED ! Method for handling field events.
CreateBox       PROCEDURE,PROTECTED ! Method to create and locate the box.
END

```

This is a fairly standard class definition that `IMPLEMENTS` one `INTERFACE`. The interface implemented is the `WindowComponent`. The seven methods declared in `WindowComponent` do not have to be declared again here – they are implicitly added to the class list of methods. However, when you create a class that `IMPLEMENTS` an `INTERFACE`, you do have to write code for every method defined in the interface, even if the method has nothing more than a `CODE` statement. More on that in a moment.

In the `PDSelectCT` class, two additional methods not defined by the interface help handle field events and manage the box to be displayed around window controls.

The `TYPE` attribute indicates that this is a generic definition – the class must be instantiated using `PDSelectCT` as the definition of the instantiated object. The `LINK` attribute adds the file to the link list; its second parameter forces it to be compiled. A little side note here is that I like to end typed class labels with the letters `CT` for `Class` and `Type` to make the code a little more readable.

The class uses its properties to track what control has focus so that the control can be colored or boxed as appropriate. The `BoxFEQ` will store the field equate of the box that will be created at run time around the currently selected field.

Class method code file header

The `pdselect.clw` header contains the `MEMBER` statement, includes the declaration file, adds equates, and declares an empty `MAP` statement – the latter is there simply to make the Clarion built-in functions available to the class methods.

```

MEMBER( )
INCLUDE('PDSelect.inc'),ONCE
INCLuDE('Equates.clw')
MAP.

```

Interface methods

As I previously stated, when implementing an interface you have to write at least minimal code for all interface methods. The example, however, leaves most of these as stubs and adds code to only two: `TakeEvent` and `SetAlerts`. `TakeEvent` calls the `TakeFieldEvent` method if the event is a field event. `SetAlerts` is called by the `WindowManager` class during initialization; I use it to color all required fields before the window opens. The LOOP using `PROP:NextField` is an easy way to loop through all the fields on a window.

Listing 1. Select Class Interface Methods Code

```

PDSelectCT.WindowComponent.SetAlerts      PROCEDURE
nextfield      LONG
CODE
LOOP
    nextfield=0{PROP:NextField,nextfield}
    IF NOT nextfield then BREAK.
    IF nextfield{PROP:Req}
        nextfield{PROP:Background}=color:red
    END
END

PDSelectCT.WindowComponent.TakeEvent      PROCEDURE!,BYTE
CODE
IF FIELD()
    SELF.TakeFieldEvent()
END
RETURN 0

PDSelectCT.WindowComponent.Kill           PROCEDURE
CODE

PDSelectCT.WindowComponent.Reset          PROCEDURE(BYTE Force)
CODE

PDSelectCT.WindowComponent.ResetRequired PROCEDURE!,BYTE
! 1 if reset of window required
CODE
RETURN 0

PDSelectCT.WindowComponent.Update         PROCEDURE
! Everything but the window!
CODE

PDSelectCT.WindowComponent.UpdateWindow  PROCEDURE
CODE

```

Non-interface class methods

The two non-interface methods added to the class color selected entry, text, and spin fields, and add a box around all fields when they are selected. The colors vary depending on the required attribute of the control.

The TakeFieldEvent method, called by the WindowComponent.TakeEvent method, sets colors and calls the CreateBox method. TakeFieldEvent sets colors if the field is found in a list of field types specified with the INLIST procedure. This method also first restores colors if the newly selected field is not the same as the last selected field.

Listing 2. Select Class Non Interface TakeFieldEvent Method Code

```

PDSelectCT.TakeFieldEvent PROCEDURE
CODE
CASE EVENT()
OF EVENT:Selected
! Capture the selected event.
! Save the field of the selected field.
SELF.CurrentField=FIELD()
! If the field has changed
IF SELF.LastField AND FIELD()<>SELF.LastField
! Restore the original color of the last field
SELF.LastField{PROP:Background}=SELF.LastColor
! Clear the last field and last color.
CLEAR(SELF.lastField)
CLEAR(SELF.LastColor)
END
! If the field is one to be colored
IF INLIST(FIELD(){PROP:Type},|
CREATE:Entry,|
CREATE:Text,|
CREATE:Spin,|
CREATE:DropCombo)
! First save the field equate and current color.
SELF.LastField=SELF.CurrentField
! Save the old background
SELF.lastColor=SELF.CurrentField{PROP:Background}
! Set the color.
SELF.CurrentField{PROP:BackGround}=|
CHOOSE(SELF.CurrentField{PROP:Req}=TRUE,034D6FEH,COLOR:Yellow)
END
! Create Rectangle for all selected controls
SELF.CreateBox()
END

```

The CreateBox method first destroys the box if one exists, and then creates a new one around the newly selected field. I found by trial and error that the box had to be destroyed rather than simply moved to work properly where there were multiple tabs on a window. I used PROP:Pixels for precise positioning. The last line of code sets the box color as a function of whether the field is required or not.

Listing 3. Non-Interface CreateBox Method Code

```

PDSelectCT.CreateBox PROCEDURE
nextfield long
lPixels  BYTE
CODE
! Destroy the last rectangle
IF SELF.BoxFEQ
    DESTROY(SELF.BoxFEQ)
END
SELF.BoxFEQ=CREATE(0,CREATE:box,|
    SELF.CurrentField{PROP:Parent})
! Set box line width
SELF.BoxFeq{PROP:LineWidth}=2
UNHIDE(SELF.BoxFEQ)
! Use pixels for position.
lpixels=0{PROP:Pixels}
0{PROP:Pixels}=TRUE
! Position the box
SETPOSITION(SELF.BoxFEQ,|
SELF.CurrentField{PROP:Xpos}-1,|
SELF.CurrentField{PROP:Ypos}-1,|
SELF.CurrentField{PROP:Width}+3,|
SELF.CurrentField{PROP:Height}+3)
! Restore the pixel status and set the color
0{PROP:Pixels}=lPixels
SELF.BoxFeq{PROP:Color}=CHOOSE(|
    SELF.CurrentField{PROP:Req}=TRUE,COLOR:RED,COLOR:gray)

```

Template implementation

You can add the class to an application with a simple global application extension template containing three #AT statements. The first adds the global include file. The second instantiates the class as an object with a one line data declaration. The third adds one line of code to add the method to the WindowManager's WindowComponent queue. More on these in a moment.

Generated code

The template should generate the code shown below: an INCLUDE statement in the global declaration area, a class declaration in the procedure declaration section, and one line of code in the ThisWindow.Init method. Note that the parameter to the AddItem method is the instantiated class name (PDSelect) and the interface name ((WindowComponent) using dot syntax.

```

! Global Declaration Section
    INCLUDE('PDSelect.inc'),ONCE
! Procedure Declaration Section (procedure with a window)

PDSelect          PDSelectCT

! WindowManager.Init code embed section

```

```
SELF.AddItem(PDSelect.WindowComponent)
```

Template

The template has the `FAMILY('ABC')` attribute, indicating that this is for ABC applications only. The procedure `#AT` statements use the `WHERE(%Window)` expression to limit population to only those procedures that contain windows.

```
#TEMPLATE(PDSelect,'PD Selected Controls'),FAMILY('ABC')
#!-----
#EXTENSION(PDSelectGlobal,'PD Selected Controls Global')↵
,APPLICATION
#DISPLAY('This extension adds colors to required fields, ↵
changes the background of selected fields, and places a ↵
board around selected fields. '),AT(,,180,64)
#!-----
#AT(%AfterGlobalIncludes)
INCLUDE('PDSelect.inc'),ONCE
#ENDAT
#!-----
#AT(%DataSection),WHERE(%Window),PRIORITY(1200)
PDSelect PDSELECTCT
#ENDAT
#!-----
#AT(%WindowManagerMethodCodeSection,'Init','()',BYTE'),↵
WHERE(%Window),PRIORITY(8501)
SELF.AddItem(PDSelect.WindowComponent)
#ENDAT
#!-----
```

Results

The result of all this work is the best kind of template: it is just one global template with no prompts, generating two lines of code in each window procedure. The `WindowComponent` interface provides a simple hook into the `WindowManager`'s queue of `WindowComponents`. Other classes can easily hook into this same component with totally different implementations – all browse classes, for example, implement `WindowComponents` that are added to the `WindowManager`.

The figures below show the effect of this implementation:

Part No:	AMTEK BLANK
Reference No:	0.0001
Description:	COVER FOR BALL
Product Code:	24
Unit Measure:	EA
Drawing:	AMTEK.BLA

Fig. 1. Required Fields Colored Red

Part No: AMTEK BLANK
 Reference No: 0.0001
 Description: COVER FOR BALL
 Product Code: 24
 Unit Measure: EA
 Drawing: AMTEK BLA

Fig. 2. Selected Required Field Colored Yellow-Orange with Red Border

Part No: AMTEK BLANK
 Reference No: 0.0001
 Description: ER FOR BALLAST
 Product Code: 24
 Unit Measure: EA
 Drawing: MTEK BLAN

Fig 3. Non-Required Field Selected Colored Yellow with Gray Border



Fig 4. Button Selected With Gray Border

Ref No	Part No	Desc	Route Dept	PC
0.0001	G	GETTING SET		99
0.0001	GD	GUNDRILL WORK		99
0.0001	GR-ACCULUBE	LUBRICANT		M
0.0001	GR-CLOTH BUFF	10X1-1/4 X 1/4 DISC		99
0.0001	GR-COMPOUND E	GRINDING POLISH		M
0.0001	GR-COMPOUND G	GRINDING COMPOUND		M
0.0001	GR-COMPOUND H	HD 30		99
0.0001	GR-FULFACEWHE	GRINDING WHEEL		99
0.0001	GR-GREASE_DAR	AP LUBAR		99
0.0001	GR-RUBBER TIRE	3" X 10" MED.HARD		99
0.0001	GR-SAIT PAD	ABRASIVE HAND PAD		99
0.0001	GUNDRILL-NEW	GUNDRILL LABOR-NEW		37
0.0001	GUNDRILL-RETIP	GUNDRILL LABOR-RETIP		38
0.0001	H	HOUSEKEEPING		99
0.0001	I	INSPECTION		99
0.0001	IF	INCANDESCENT FIXTURE		13
0.0001	INCAND. FIXT.	CONSISTING OF:		13
0.0001	INSTALLATION	INSTALLATION CHARGE		28
0.0001	INTERIOR LITES	SEPTA MARKER-FRANK		13

Fig 5. List Box Selected With Gray Border

Tabulate Drawing
 Obsolete

Fig 6. Check Box Selected With Gray Border

With the code in a class container, maintainance is easy. Without changing the template, you could, for example, make make the class user configurable by adding color properties and a preference window called by an alert key. The simple example here, however, more than amply illustrates the purpose – interfacing with an interface.

[Download the source](#)

Philip S. Will is President of [ProDomus, Inc.](#), a SoftVelocity Third Party Accessories Partner and Clarion applications developer. He has been a presenter at several Clarion conferences, and has published articles on Internationalization and template writing. His principal third party products include PD Lookups, PD Translator Plus, and PD I-Touch Date Tools. Philip has been coding in Clarion since 1991.

Reader Comments

[Add a comment](#)

Great article Phil. I've played with this before -- your...

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](#) > [DevCons](#) > [ETC III](#)

ETC III: Various Photos

Published 2002-06-13

These are thumbnail images of the full-sized pictures. To view a picture, or to add your own comments to a picture, click on the thumbnail. Any comments you add will appear on the [Reader Comments](#) page along with a link to the image (unless you specify a private message to the editor). Thanks to Bruce Johnson, Brad Noble, Nik Johnson, and Brice Schagane for their contributions.





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

Data Structures and Algorithms Part III - Maintaining Order

by Alison Neal

Published 2002-06-19

In my [previous article](#) I introduced a data structure known as the Linked List. The Linked List is an extremely efficient way of storing and manipulating large quantities of data in memory. However, in its typical form it provides no way of sorting that data. Think of the browse templates in Clarion and how it is possible to display data by key order or the Clarion Queue structure that has the handy SORT function built in. With the List, the data can be sorted in one of two ways, either by maintaining the data in order when each element is added or by using a sort algorithm after the data has been added. In this article, I will outline how to implement a data structure known as an Ordered Link List. I'll also cover a few of the better known sort algorithms.

The Ordered Linked List, unlike the standard Linked List, contains clearly defined rules of precedence in the Add method, so that all items in the list are maintained in correct order when inserted. Unfortunately, it's not always possible to have clearly defined rules of precedence in advance of adding the data, and that's why it's also useful to know a few sort algorithms. The algorithms I will cover in this series include:

- **Bubble Sort**
- **Cocktail Shaker Sort**
- **Counting Sort**
- **Heap Sort**
- **Insertion Sort**
- **Merge Sort**
- **Quick Sort**
- **Radix Sort**
- **Selection Sort**
- **Shell Sort**

This may seem like a lot of algorithms to cover just to sort some data, but some of these algorithms can be very enlightening, and offer insights in how to solve diverse problems.

The Ordered List

The Ordered List differs from the Linked List I covered [last month](#) as it includes clearly defined *rules of precedence*. These rules say which order any given data should be placed in, whether descending or ascending. Implementing these rules required a complete rewrite of the Add methods. The Linked List covered in my [previous article](#) included `AddBefore` and `AddAfter` methods; the Ordered List relies on rules of precedence so these methods are no longer required. Instead I've used the following code:

```
OList.OLAdd          PROCEDURE(ULONG YourVal)
  CODE
  SELF.head &= SELF.OAdd(YourVal, SELF.head)

OList.OAdd          PROCEDURE(ULONG YourVal, *listNode t)
  CODE
  IF t &= NULL OR t.nodeVal >= YourVal
    SELF.Curr &= NEW(listNode)
  ?  ASSERT(~SELF.Curr &= NULL)
    SELF.curr.nodeVal = YourVal
    SELF.curr.nextNode &= t
    SELF.count += 1
    RETURN SELF.Curr
  ELSE
    t.nextNode &= SELF.OAdd(YourVal, t.nextNode)
  END
  RETURN t
```

So, what's happening here? First I call `OLAdd`, passing to it the value that I want to add to my list. Then `OAdd` is called passing the required value and the starting point of the list. This allows me to use a recursive function for the `OAdd` method.

Consider what happens if I add the following numbers: 12,22,1,5. First I pass the value 12 to the `OLAdd` method. As the list should have just been initialized, `SELF.head` should be `NULL`, so when the `OAdd` method is called the `IF` statement should execute. Memory is allocated for the new node and the value is assigned, so that on returning `SELF.head` should equal `12 -> NULL` (the arrow indicates the link to the next node).

On the second call to `OLAdd` I pass the value 22. `OLAdd` then calls `OAdd`, passing the head of the list. This time, however, the `IF` statement will not execute, as the head value 12 is less than 22 and `t` is not `NULL`, so the method calls itself passing `t->nextNode` which is `NULL`. This time the `IF` statement will execute and allocate the appropriate memory. The method then returns a list containing `22 -> NULL`, which is assigned to the end of the original value of `t` which was `12 ->`, thus making a now complete list of `12 -> 22 -> NULL`.

On the third call to `OLAdd` I pass the value 1, which is lower than the head node of 12 which was originally passed to the `OAdd` method. Now memory is allocated for the new node, and what was previously allocated is linked to the new node, so the list now looks like this: 1 -> 12 -> 22 -> NULL.

On the fourth call to `OLAdd` I pass the value 5. This is not lower than the head node, and so the `OAdd` method recurses once to add the new node with the value 5. The List eventually ends up being 1 -> 5 -> 12 -> 22 -> NULL.

In the sample code available at the bottom of this article, you may find that other changes have been made to the List class, including new remove functions.

Sort algorithms

Depending on what you're trying to achieve with your code, it isn't always possible to implement clearly defined rules of precedence, and I've frequently found that I need to sort data in more than one way. Because of this, the ability to sort data in an efficient manner has become a very useful tool for me. In the downloadable code (available at the bottom of this page) I have taken a step backwards and coded the different sort algorithms for arrays. This will hopefully get people familiar with the idea of the algorithms prior to having reference variables involved. However, please remember that sort algorithms can become significantly more efficient when using structures such as the list, because these structures only require the assignment of reference variables rather than the assignment of values stored in memory, which is what happens with an array.

The sort algorithms that I will introduce to you are, in most cases, sufficient for sorting small quantities of numbers. In situations where you are dealing with truly significant amounts of data, you may want to design and develop a Combo Sort Algorithm, which is a sort algorithm which is made up of a combination of two or more other sort algorithms. For example, in my experience you do not want to use a Bubble Sort to sort 50,000 or more numbers, unless you're quite happy with overnight processing. Comparatively, the Shell Sort (more about that at a later date) will handle 50,000 numbers with remarkable ease, but can be inefficient when dealing with only a small amount of data.

Swap Procedure

It's important to learn how to perform a basic swap, as this is the foundation of many sorting algorithms::

```
temp = A
A = B
B = temp
```

Variable A and B are passed to the procedure by address, because these are the variables that I want to swap. Temp is a local variable that is used just to store the temporary value A. before A is overwritten with the value stored in B. So, if I want to swap A = 6 with B = 3:

```
temp = 6
A = 3
B = 6
```

I've separated the swap function in my code because I've found that a lot of sort algorithms use it to move values into their appropriate positions, and it saves me from having to maintain duplicate code within those algorithms. The Bubble Sort algorithm outlined below, and the Cocktail Shaker Sort I'll discuss next week, both use the swap function.

Bubble Sort

The Bubble Sort is the most common of the sort algorithms. I learned this algorithm 12 years ago when I studied Cobol, and found it was still being taught in Computer Science ten years later. When you need to sort a small amount of data, this algorithm is all you need:

```
LOOP i = 1 TO SZ - 1
  LOOP j = 1 TO SZ - i
    IF myArray[j] > myArray[j+1]
      swap(myArray[j], myArray[j+1])
    END
  END
END
```

Hopefully you can already see why you don't want to perform this sort on over 50,000 numbers. How many iterations will the nested loop go through if the size (SZ) of the array is 50,000? How many comparisons are going to be performed? This algorithm checks every item individually, and then it checks it again, and again, and again, "bubbling" the highest values to the top.

For example, consider an array which contains the following values: 22, 99, 4, 12, 33, 9. The size of the array is 6. The first loop is executed 5 (Size - 1) times, the nested loop is executed 5 times in the first iteration, 4 in the second iteration, 3 in the third etc.

Within the nested loop a comparison is performed as to whether the current value is greater than the next value. You will note that this is why the nested loop only executes from 1 up to Size - i times, as this ensures that the algorithm does not overrun the bounds of the array. If the current value is greater than the next value, then the two values are swapped.

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

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

Figure 1. The bubble sort, with swapped numbers in red.

The excessive number of comparisons that can occur in the bubble sort can seriously affect processing time. Looking at Figure 1, you should be able to see how the number 99 bubbles its way up to the last slot of the array. The second time the nested loop is entered; the value 22 is bubbled up to a better position. Notice also, as with most algorithms the Bubble Sort will continue until the iterations are completed. It will not stop processing when the array is sorted, as it has no way of telling whether the array is sorted or not.

An interesting test of speed for a Sort Algorithm is to pass an already sorted array to it. With the bubble sort the time taken to sort an already sorted array and an unsorted array are about the same, with the sorted array being only slightly quicker. On my PC, for an array of 2,500 values randomly generated in the range of 1 – 999, the speed difference was minimal (a click is 1/100 of a second):

Sort Name	Unsorted Array		Sorted Array	
	clicks	clicks	clicks	clicks
Bubble Sort	824	clicks	650	clicks

The Bubble sort a good example of basic sorting, but there are many other ways to sort lists. [Next week](#) I'll look at the Cocktail Shaker, Insertion, and Counting sort algorithms.

[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](#)

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](#) > [Browses](#) > [Browses, Using](#)

The Checkbox Class Performance Upgrade

by **David Harms**

Published 2002-06-20

Last year I wrote an [article](#) about using checkboxes to manage many-to-many relationships between files. At the time I was developing the Clarion Magazine topical index, and I wanted an easy way to associate articles with topics. The interface I eventually settled on was a left-side browse box showing the articles, and a right-side browse box showing a list of topics. The first column of the right-side list showed a checkbox, which I could check/uncheck simply by clicking once on the first column. Figure 1 demonstrates the concept, although it isn't a shot of my program, but of the Enroll application which is my demonstration vehicle for the checkbox code.

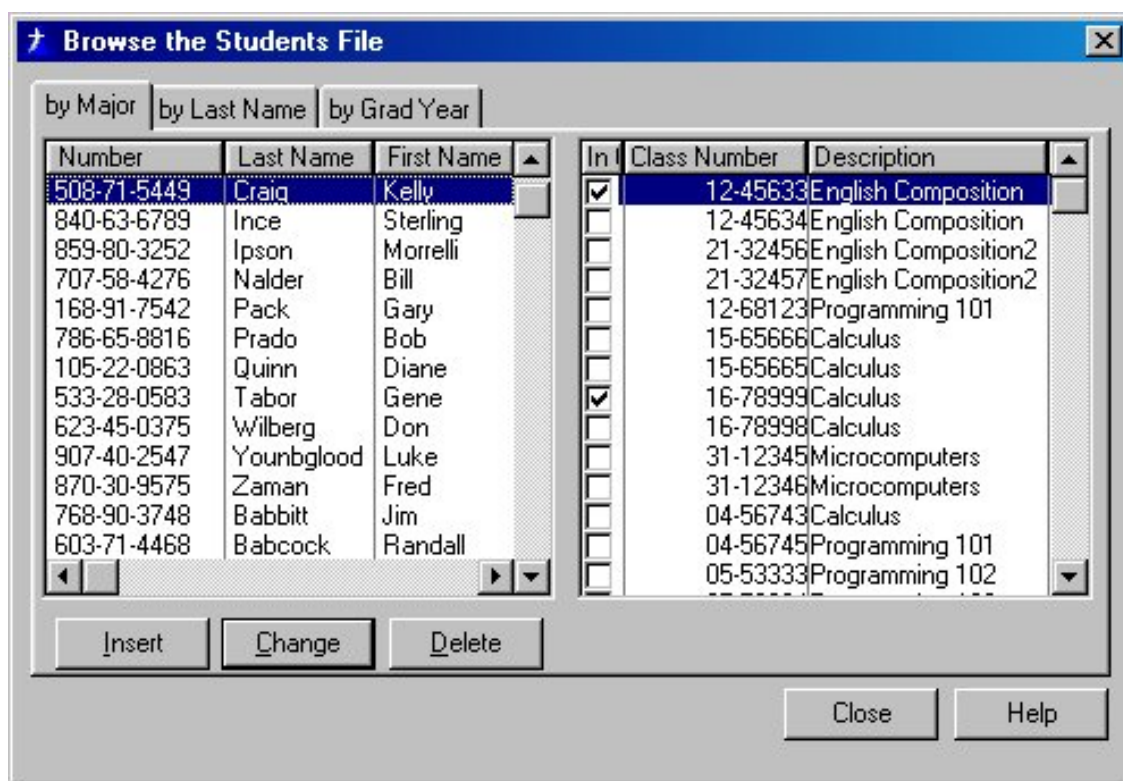


Figure 1. The Enroll application, using checkboxes to associate records.

I was quite happy with the results for quite a while. And then a couple of months ago, following the birth of our daughter, I began working mostly from home. That meant that I no longer had a LAN connection to the Clarion Magazine MySQL database - instead, I was connecting via cable modem. And while performance on most browses was fairly snappy, those which used my checkbox class were dog slow. In fact they were almost completely unusable, to the point where it became difficult to maintain the topical index.

The problem was that although the solution I arrived at was relatively easy to implement, and leveraged ABC code, it involved a lot of network traffic. I let the browse class determine which checkboxes to check, and it did that by using a JOIN between the table with the list of available options, and the table with the linking data.

Every time I selected a new article in the left side browse, the right side browse with the checkboxes had to be completely rebuilt (triggered by a “reset” field) to get the new data (see my earlier article for a full description of my arbitrary meanings for “left” and “right” side browses in the context of the checkbox code). I didn’t notice the performance hit on a local 100Mb network connection, but it was readily apparent on a cable modem connection, with an effective throughput of something like 500Kb.

After a little analysis, I realized that I didn’t need to completely rebuild that right-side list box at all. In normal day-to-day use, the list of topics didn’t change, so I only needed to read that information once, when the procedure opened. All that really did change was the checked records, and usually only a few of these were checked for any one article. What I needed to do was get the linking data alone whenever the left browse changed, and use that information to update the right browse checkboxes. Also, to avoid a hit on scrolling the right side list box, I decided to file load the browse (putting all the data in memory) instead of page load. Again, I never had more than a few pages of data and initial load penalty was minimal.

The end result was an updated browse class that showed excellent performance across the ‘net. Here’s how it works.

The declarations

I won’t reiterate the concept behind the checkbox browse approach – for that, please read my earlier article. Also see that article for additional background on implementing the class in your apps. I’ll just cover the highlights here.

This is the class declaration:

```
cciBrowseClassB CLASS(BrowseClass),TYPE,MODULE('ccibrowb.clw')|
```



```

    ,LINK('ccibrowb.clw',_ABCLinkMode_),DLL(_ABCDllMode_)
LastTime      long,protected
DebugQ        &queue
Debug         byte(0)
DataQ         &DataQ,protected
ViewQIconField &long,protected
RightPrimaryID &long,protected
LinkLeftField &long,protected
LinkLeftFieldName cstring(255),protected
LinkRightField &long,protected
LinkKey       &key
LeftPrimaryID &long,protected
lc           long,protected
LinkFM       &FileManager,protected
ViewQ        &queue,protected
ViewQIconField any
ViewQRightField &long
!--- methods ---
DebugMsg     procedure(String msg)
Init         procedure(FileManager LinkFM,Key LinkKey|
                    ,*long LinkLeftField,*long LinkRightField|
                    ,*long LeftPrimaryID,*long RightPrimaryID|
                    ,Queue ViewQ,*? ViewQIconField|
                    ,*long ViewQRightField)

Init         procedure(cciBrowseClassBParams params)
Kill         procedure,virtual
LoadCheckboxData procedure
RedisplayRecord procedure
SaveCheckboxData procedure
SetIcons     procedure
TakeEvent    PROCEDURE,virtual

end

```

cciBrowseClassB bears a lot of resemblance to the original cciBrowseClass. It adds a couple of new properties, including ViewQ, ViewQIconField, and ViewQRightField. ViewQ is the generated queue definition which the browse box uses to display items, and which is manipulated by the ABC browse object. So why don't I just get the view queue from the browse? For one thing, it's a private property buried in the browse internals. I could grab it by overriding the browse INIT method, but I need two fields from that queue anyway, so I decided to include it in my INIT call. More about those fields in just a bit.

You'll notice that there are now two INIT methods. I've left the original style INIT, with many parameters, but the one I use now is the second version, which takes a GROUP structure as a parameter. This is prototyped as follows:

```

cciBrowseClassBParams  group,type
LinkFM                 &FileManager
LinkKey                &Key
LinkLeftField          &long
LinkLeftFieldName      string(255)
LinkRightField         &long

```

```

LeftPrimaryID          &long
RightPrimaryID         &long
ViewQ                  &queue
ViewQIconField         any
ViewQRightField        &long
                        end

```

The only point of interest in this declaration is that ViewQIconField is declared as an ANY variable. This is because in Clarion 5.0 the queue field that stores the icon setting is a SHORT, while in C55 it's a LONG. By using the ANY I can store a reference to either data type. I don't use ANYs discriminately, however, because they are slightly more expensive than simple data types.

In the example application I've declared an instance of this group in the procedure data section (although it could also go in the global data section):

```

BrowseParams           group(cciBrowseClassBParams)
                        end

```

In the WindowManager.Init embed, priority 8800, I included the following initialization code:

```

BrowseParams.LinkFM           &= access:Enrollment
BrowseParams.LinkKey          &= ENR:StuSeq
BrowseParams.LinkLeftField    &= ENR:StudentNumber
BrowseParams.LinkRightField   &= ENR:ClassNumber
BrowseParams.LeftPrimaryID    &= STU:Number
BrowseParams.RightPrimaryID   &= CLA:ClassNumber
BrowseParams.ViewQ            &= Queue:Browse
BrowseParams.ViewQIconField   &= Queue:Browse.InClass_Icon
BrowseParams.ViewQRightField  &= Queue:Browse.CLA:ClassNumber
ClassesBrowse.Init(BrowseParams)

```

The original, single line way of calling the INIT looks like this:

```

ClassesBrowse.Init(
    access:Enrollment,      | ! Linking FileManager
    ENR:StuSeq,             | ! Linking File key
    ENR:StudentNumber,     | ! Linking File left field
    ENR:ClassNumber,       | ! Linking file right field
    STU:Number,            | ! Left file primary key field
    CLA:ClassNumber,       | ! Right file primary key field
    Queue:Browse,          | ! View queue
    Queue:Browse.InClass_Icon, | ! Icon field
    Queue:Browse.CLA:ClassNumber) | ! Queue right field

```

The same variables are passed in the same order in both cases. I find the first approach easier to deal with when using long parameter lists. As long as I use reasonably descriptive field names, I can dispense with the comments – the field names should be information enough. It's also easy for the compiler to tell me if I've made an incorrect assignment. The downside is that

if I forget to initialize one of the group members, the compiler won't care.

The multiple-parameter call is a bit trickier to get right, in my experience. The sole benefit that I can see (aside from a possible miniscule performance improvement due to creating less temporary data) is that it will tell me when I forget to include a parameter. But it's easier to get the order of parameters wrong, unless the call is well-commented. In any case, you have your choice, since the INIT that takes a group as a parameter just calls the non-group INIT with the group's data.

As in my [previous article](#), I've modified the Clarion5 Enroll application to demonstrate the use of the class.

The code

The original version of my checkbox class has two methods that do all the work: `SetQueueRecord`, and `TakeEvent`, both of which are virtual methods derived from the `ABC BrowseClass`. `TakeEvent` is automatically called when the user clicks on a checkbox on the right hand browse, and updates the linking data accordingly, inserting or deleting a record. `SetQueueRecord` is automatically called when the browse wants to assemble the data for one line in the list box. Since I wasn't going to rely on the rebuilding of the right hand browse data any more, I couldn't use `SetQueueRecord`. That meant I had to find another way of loading the checkboxes when the user selected a new left browse record.

I added a method to the class called `LoadCheckboxData`, and placed a call to this method in the left side browse's `TakeNewSelection` method, after the parent call. In the example app, the code looks like this:

```
ClassesBrowse.LoadCheckboxData()
```

Here's the source for that method:

```
cciBrowseClassB.LoadCheckboxData      procedure
x      long
      code
      free(self.DataQ)
      self.LinkRightField = 0
      self.LinkLeftField = self.LeftPrimaryID
      set(self.LinkKey,self.LinkKey)
      loop while self.LinkFM.next() = level:benign
          if self.LinkLeftField <> self.LeftPrimaryID
              break
          end
      self.DataQ.ID = self.LinkRightField
      self.DataQ.CurrValue = 1
      self.DataQ.Changed = false
      add(self.DataQ,self.DataQ.ID)
```

```

end
self.SetIcons()

```

The class contains a queue of linking data information declared as follows:

```

dataQ      queue, type
ID         long
OrigValue  long
CurrValue  long
          end

```

Any time `LoadCheckboxData` is called, that queue is cleared. The method then looks for linking data related to the current left side browse, and stores a corresponding entry for each record found in `DataQ`. Once that process is complete, `LoadCheckboxData` calls the `SetIcons` method:

```

cciBrowseClassB.SetIcons PROCEDURE()
x      long
  code
  loop x = 1 to records(self.ViewQ)
    get(self.ViewQ,x)
    self.DataQ.ID = self.ViewQRightField
    get(self.DataQ,self.DataQ.ID)
    if errorcode()
      self.ViewQIconField = 1
    else
      self.ViewQIconField = choose(self.DataQ.CurrValue=1,2,1)
    end
    put(self.ViewQ)
  end
end

```

The `SetIcons` method simply loops through the queue that the browse uses as its source of data to display, and updates the icon field so that a checked or unchecked icon displays properly.

There are a couple of points worth noting here. First, to understand how this works you must have already set up an icon field in the browse, and that you've created the icons specifically as instructed in the original article. This is because the `CHOOSE` statement will assign a value of 1 or 2 to the icon field, and if you've got it backwards, all your unchecked boxes will indicate linking data, and vice versa.

The second point is that this class is a lot closer to the metal than the first class, which hooked in at a higher level of abstraction in the browse display process. This class does in fact need a reference to the browse display queue, to the queue field that stores icon display information, and to the queue field that stores the linking ID. Once again, here are the relevant fields passed to the `INIT` method:

```

BrowseParams.ViewQ           &= Queue:Browse
BrowseParams.ViewQIconField  &= Queue:Browse.InClass_Icon
BrowseParams.ViewQRightField &= Queue:Browse.CLA:ClassNumber

```

This is the generated Queue:Browse declaration:

```

Queue:Browse      QUEUE
InClass           LIKE(InClass)
InClass_Icon     LONG
CLA:ClassNumber  LIKE(CLA:ClassNumber)
COU:Description  LIKE(COU:Description)
CLA:ScheduledTime LIKE(CLA:ScheduledTime)
COU:Number       LIKE(COU:Number)
Mark             BYTE
ViewPosition     STRING(1024)
                END

```

Note that the field I'm passing for the icon information isn't the one corresponding to the local field you defined as the icon field. In fact, that contents of that field are utterly irrelevant. It's there so that the LONG (or in C5, SHORT) field after it will be created. The contents of that field, in conjunction with the generated calls to PROP:IconList at procedure setup, determine the icon that's shown:

```

?List{PROP:IconList,1} = '~off.ico'
?List{PROP:IconList,2} = '~on.ico'

```

If I place a value of 1 in Queue:Browse.InClass_Icon, off.ico will be displayed; a value of 2 shows on.ico.

Again, because I'm not reloading the right browse's data each time the left browse selection changes, I need to work directly with the browse's queue. The field Queue:Browse.CLA:ClassNumber, together with the student number from the left browse, is what I need to look up the Enrollments data which has been saved in DataQ.

The next method of interest is TakeEvent. In the original, TakeEvent retrieved the current enrollment record, updated it, and then triggered a redisplay of that record. This version, like SetIcons, works a bit more directly with the browse queue.

```

cciBrowseClassB.TakeEvent PROCEDURE
code
parent.TakeEvent()
if field() = self.lc
  if event() = event:Accepted |
    and keycode() = MouseLeft |
    and self.lc{proplist:mouseuprow} |
    = self.lc{proplist:mousedownrow} |
    and self.lc{proplist:mouseupfield} |
    = self.lc{proplist:mousedownfield} |
    and self.lc{proplist:mousedownfield} = 1

```

```

    get(self.ViewQ,choice(self.lc))
    self.DataQ.ID = self.ViewQRightField
    get(self.DataQ,self.DataQ.ID)
    if errorcode()
        self.DataQ.ID = self.RightPrimaryID
        self.DataQ.CurrValue = 1
        self.DataQ.Changed = true
        add(self.DataQ,self.DataQ.ID)
    else
        self.DataQ.CurrValue |
            = choose(self.DataQ.CurrValue = 1,0,1)
        self.debugmsg('setting to ' & self.DataQ.CurrValue)
        self.DataQ.Changed = true
        put(self.DataQ)
    end
    self.setIcon()
    self.SaveCheckboxData()
end
end
end

```

First, TakeEvent determines that the user has clicked on the first column, which is where the checkbox is placed. It then grabs the data queue record corresponding to the linking field in the right browse, and flips CurrValue, which is either 1 or 0, depending on whether the checkbox is checked or unchecked.

DataQ records accumulate as you check new records, but don't disappear when you uncheck a record. That's because you still need to know if you have to delete a linking record.

The SaveCheckboxData method is, in this implementation, called each time you click a checkbox, but it's designed so that can batch updates when the left side browse selection changes. You just have to work out where to call the method (i.e., that's an exercise left to the reader). Here's the method code:

```

cciBrowseClassB.SaveCheckboxData      procedure
select  cstring(500)
x      long
code
loop x = 1 to records(self.DataQ)
    get(self.DataQ,x)
    if (self.DataQ.Changed)
        if self.DataQ.CurrValue = 0
            ! If the link exists, remove it
            self.LinkLeftField = self.LeftPrimaryID
            self.LinkRightField = self.RightPrimaryID
            if self.LinkFM.Fetch(self.LinkKey) = level:benign
                compile('***',_c55_)
                self.linkFM.DeleteRecord(0)
                ***
            omit('***',_c55_)
            delete(self.LinkFM.File)
            ***
        end if
    end if
end loop
end

```

```

        end
    else
        ! Create the link
        self.LinkLeftField = self.LeftPrimaryID
        self.LinkRightField = self.RightPrimaryID
        self.LinkFM.TryInsert()
    end
    self.DataQ.Changed = false
    put(self.DataQ)
end
end
end

```

The code is straightforward – it simply loops through DataQ. If the CurrValue of the DataQ record is 1, it attempts to create the record. This code assumes a unique key on the linking values, or you could get multiple linking entries created. If CurrValue is 0, it tries to fetch the record, and if successful, deletes the record.

Upgrading

If you're upgrading from the previous version of this class, you'll need to make sure you take the following steps, in addition to changing the INIT call.

- Remove the Reset field from the right-hand browse. You don't want this browse refreshing every time.
- Set the right hand browse's load method to File instead of Page.
- Change the right hand browse class from cciBrowseClass to cciBrowseClassB (you may need to refresh the list of ABC classes)

The only downside I've discovered so far to the approach I've discussed here has to do with page loading. Because the right hand browse is never reset, I have to restart the procedure if I want any changes to the base data for that browse to appear.

Summary

The first iteration of my checkbox management class had the advantage of a high level of abstraction, and made the most of existing ABC code. Unfortunately, it resulted in way too much bandwidth to be usable across an Internet connection. This version of the class keeps the amount of data retrieved from the server to a bare minimum, and results in excellent performance over a relatively slow database connection.

[Download the source](#)

[David Harms](#) is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with

with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995). His most recent book is [JSP, Servlets, and MySQL](#), published by HungryMinds Inc. (2001).

Reader Comments

[Add a comment](#)

I've updated the source, with the following code added to...

Please Steffen Rasmussen could you update your template...

Correction - I meant to put the code self.ViewQIconField...

An updated template from Steffen is on the way...

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

[SetupBuilder SetupScript FAQ](#)

Friedrich Linder answers some commonly-asked questions about SetupScript in this posting on the SoftVelocity news server.

Posted Wednesday, June 26, 2002

[XPMenu Lookalike Developments And Holiday Schedule](#)

Ronald van Raaphorst will be on holiday during the first three weeks of July, so there's no support during this time. Upcoming improvements to XPMenu include: Optionally let the end user edit/add/delete menu items, which can be used lists you use; An item can now call a procedure, just like a button can do. The template is also Clarion5EE B compatible for ABC templates.

Posted Monday, June 24, 2002

[SysTree 1.1 Released](#)

SysTree 1.1 is now available. This version contains some minor improvements and bugfixes. A complete list of changes can be found in the documentation. The installation password has not changed. SysTree is a wrapper class for the tree-view common control, allowing Clarion apps to use tree-view controls with the real Windows look and feel. Also contains SysDirTree, a derived class that allows browsing through file system folders just like the Windows Explorer does. Demo available.

Posted Monday, June 24, 2002

[CapeSoft MessageBox 1.6d](#)

A general update to MessageBox is now available.

Posted Monday, June 24, 2002

[Secwin 3.16d Beta 6](#)

A minor update to the SecWin beta is now available.

Posted Monday, June 24, 2002

TickerTape 1.61

TickerTape has a minor update - it has been converted to SAF format.

Posted Monday, June 24, 2002

Insight Graphing 1.0 beta 7g

The latest beta of Insight Graphing has more bug fixes.

Posted Monday, June 24, 2002

HyperActive 1.7 Update

HyperActive 1.7 now only displays items as hyperlinkable if a registered program for that file-type exists.

Posted Monday, June 24, 2002

Special Agent 1.41

A bug fix update to Special Agent is now available.

Posted Monday, June 24, 2002

File Explorer 2.4c

A general update to File Explorer is now available.

Posted Monday, June 24, 2002

File Manager 3 Limited Release

The first alpha release of the new File Manager 3 project is now available. This version supports auto-upgrading of MsSQL databases. As this is based on some very new technology CapeSoft is inviting users to take part in a free alpha testing, but only a few will be selected. After the alpha phase normal public beta testing will proceed. Of particular interest is a feature which converts data from existing TPS and Clarion DAT files into MsSQL tables. Interested testers please email.

Posted Monday, June 24, 2002

Programming in Clarion's ABC - Book Release

After Bruce's session at ETC 3 this year got such overwhelming feedback, CapeSoft has been press-ganged into making the session notes available to the wider Clarion community.

Described as a "work in progress" these notes currently cover about 140 pages. Items covered include a Tutorial section which explains how ABC works, detailed chapters on the File Manager and ErrorClass classes, and various other parts of ABC, like Popups, IniManagers and Edit in Place. The book costs \$50 plus shipping. At the moment the book is available in a printed format only.

Posted Monday, June 24, 2002

[xAccess Manager Lite v1.2](#)

xAccess Manager Lite is a library with set of extension and code templates for fast and easy installation of an access control capabilities. Just insert the global extension template into your application. xAccess Manager Lite is based on the list of the users, list of procedures (Access Objects), and relations between them. Access Manager Lite supports the "standard" browse-form paradigm. Supports single EXE, multi Dll (Local Mode, Standalone Mode), 32-bit.

Posted Monday, June 24, 2002

[Imaging Templates Version 1.14 Released](#)

NextAge has released version 1.14 of the Imaging Templates. This is the template chain that uses the Windows Imaging OCXs. Changes with this version include: Fixed FileDialog error when using ExportButton; Added PageProperties Control Template; Added ImageSettings (Procedure to get Image Settings) Procedure Template; Added NA::MarkSelect Gets the information about the selected Annotation during the MarkSelect event; Posting EVENT:NA_MarkSelect when the MarkSelect event occurs; Added NA::SelectRect Gets the information about the drawn rectangle during the SelectionRectDrawn event; Posting EVENT:NA_SelectionRectDrawn when the SelectionRectDrawn event occurs.

Posted Monday, June 24, 2002

[INN Bio For 20-June-2002](#)

After a weeks-long break, the bios are back! And they resume with a well-known Clarionite, usually identified on the newsgroups by his car... but the car in his bio is gold, not purple... hmmm. Originally from India, now at home in New Jersey, he works at being a programmer and "Mr. Mom." With one adorable son and another little one on the way, we know he stays busy! And he's included some great photos, too.

Posted Monday, June 24, 2002

[EmailReport 2.5 Released](#)

The EmailReport 2.5 templates (both ABC and Legacy) are now available from ClarionShop. New features include: Email any Clarion, CPCS, RPM, DAS, and Tintools report right in the body of your email; Populate email addresses, Subject and Body note right from your Clarion fields; Duplicate report as RTF attachment (which can be opened even with Windows WordPad (95-XP)) ; Use template just for saving reports to disk in this universal RTF format; Play sound files in your emails.

Posted Monday, June 24, 2002

[xWhatsNew v1.1 Released](#)

New in xWhatsNew v1.1: New type of history line: % Bugfix; Legend for history files in the top of the file. Remember to import new xWhatsNew procedure from xWN.TXA. New install

and demo available.

Posted Monday, June 24, 2002

[Minor Update To FSDateS Class](#)

Tony York has updated the fsdates class to fix a GPF caused by an out of scope variable.

Posted Monday, June 24, 2002

[EmailData 1.3 Released](#)

A new version of Vivid Help's EmailData template is now available for download. If you are current user, use your password to unzip the file. A shareware version is coming. New features include: Populate email Subject, Body and Attachments directly from you Clarion fields; Automatically embed images into the body of you emails; Send and play sound files (*.wav, *.mid, *.ram, *.ra, *.aif, *.au) with your emails.

Posted Monday, June 24, 2002

[Clarion Third Party Profile Exchange Updated](#)

The Clarion Third Party Profile Exchange consists primarily of profiles of third party add-on products and vendors. This includes freeware templates and tools as well. Online and Downloadable Profiles available. Online product profiles include Product Internet URL, Order URL, Dated Price Quote, Grouped by Category, Extended Description and download Page Reference! Currently, there are 378 product profiles and 319 vendor profiles. You must have the recently released Product Scope 32 PRO Version 4.5 to view profiles with data files.

Posted Thursday, June 20, 2002

[Jack Toering On Network Operating Systems](#)

In Jack's own words, " It seems I do a big article on NOSes ... about every two years, so I guess it's time." This epic newsgroup posting is jammed full of information and opinion.

Posted Wednesday, June 19, 2002

[Scripting Language For C55](#)

Fabian Szenkier of Afiq Information Systems has posted some samples and internal documentation for a Clarion 5.5 scripting language. If there are more than 20 interested customers of Power Interpreter, AIS will sell it for \$499 per unit.

Posted Wednesday, June 19, 2002

[Latest D2MySQL Templates And Example](#)

Roberto Artigas has posted an updated version of his dictionary-to-MySQL templates, including the C5.5G Autolog example app converted to MySQL.

Posted Wednesday, June 19, 2002

Coollook Template Demo

A demo Avi file (executable, 6.73MB) is now available. This demo shows how to add and configure the Coollook template.

Posted Tuesday, June 18, 2002

chSTD Library v2.6 Released

The chSTD library consists of numerous modules and functions, including: business functions (accounting, analytical geometry (coordinate geometry), inverse hyperbolic, hyperbolic, matrix algebra, mathematical, statistical, trigonometric functions), handling of constants, functions with the expanded possibilities, basic classes, classes with the expanded possibilities, services, screen services, standard components. Each module includes hundreds functions, procedures, components, classes and services. Clarion 5.5 or later, ABC, Legacy (hand code), 32-bit only. The business functions module - chFM is now available. Libraries, templates, examples, and complete documentation included.

Posted Tuesday, June 18, 2002

EasyReport v1.01 Released

EasyReport 1.01 is now available. New features include: Text controls resizing and adjustment of the vertical lines and boxes height according to the text control's height; Greenbar effect; Splitting large text between pages with vertical lines and boxes adjustment; Setting number of report copies; Printing page of pages and total number of pages; Compatibility with CPCS templates. EasyReport works with Clarion 5 and Clarion 5.5, both ABC and Legacy Templates

Posted Tuesday, June 18, 2002

Arnor & Sue In Iceland

Arnor has posted the first of several articles about his and Sue's (and Laila's) trip to Iceland. He will be posting the rest of the story over the next month or so.

Posted Tuesday, June 18, 2002

Freeware Reports-To-WMF Template Updated

Version 2.0 of TOGRAPH has been released. The template (no DLL or LIB anymore) will create numbered WMFs from your reports. This freeware was kindly supplied by Alex Oberhard of Komputery (Poland). Compatible with Clarion 5 and 5.5 ABC

Posted Tuesday, June 18, 2002

Clarion/ASP Support And Training Newsgroups Online

The Clarion/ASP Support and Training groups are online and ready for your use. A personal email will be sent out shortly to each purchaser of the Clarion/ASP templates with details about the training schedule. Your account information and the windows client software is included on your CD. Please direct all support related posts to discuss.softvelocity.com.

Posted Tuesday, June 18, 2002

[**XPMenu Extra Demo**](#)

A new demo is available for XPMenu. Instead of the BrowseBoxUpdateControl buttons, there is a file actions menu (which can be used for several browse boxes, depending on which browse box has focus). A not implemented feature is to hide/unhide a complete menu when a control is or is not visible (no handcoding needed). The application also has a menu on which the end-user can add, rename or delete items. In this demo, the items are views (i.e. column sets) for the browse box. Just right click the 'columns' menu and choose your liking.

Posted Tuesday, June 18, 2002

[**Freeware: Create XLS Files In Clarion**](#)

Klarisoft Inc. has released ExcelWrite, a class that writes an Excel BIFF 2.1 spreadsheet file directly - no external DLLs or Excel OLE automation needed. Everything is pure Clarion source. Great for exporting data from your application to an Excel spreadsheet without using any 3rd party tools. Features include: ; Set individual cell values and value types; Set the spreadsheet margins in inches; Set individual or a range of column widths ; You can specify the font, alignment and formatting for individual cells; Handles cell borders, shading, locking and hiding ; You can specify Headers and Footers to appear on each printed page ; You can specify whether or not to print GridLines ; Individual cells can be protected; Horizontal Page Breaks; Default (global) row heights that affect the entire spreadsheet ; Ability to set the height of individual rows. ExcelWrite Class is Freeware. You may freely use it in your Clarion applications. No payment is required for the use of this class but a donation would be appreciated.

Posted Tuesday, June 18, 2002

[**Clarion/ASP Video**](#)

SoftVelocity has two short video demos of its Clarion/ASP product. Part 1: How quickly can you build and deploy a Clarion/ASP application, and how much do you need to know about ASP? Learn the answers in less than four minutes. Part 2: What can you do with just a little customization? Here's what you can do with a product that completely separates the user interface from the application logic. Approximately five minutes.

Posted Thursday, June 13, 2002

[**Bruce Johnson's ABC Notes Now Available**](#)

Recently Bruce Johnson presented an all-day training session at ETC 3. Recent demand for the notes, from people who were unable to attend the session, has been overwhelming. CapeSoft has made the notes available, in a printed format, to the general Clarion community. Cost is \$50 plus shipping.

Posted Wednesday, June 12, 2002

[Free XML Wrapper](#)

Plugware Solutions and ThinkData have released the MS XML wrapper as a free product. After you download the free COM classes, you'll see a link for the XML wrapper. You must have the COM classes installed and your redirection file edited properly for the XML wrapper to be useful.

Posted Wednesday, June 12, 2002

[SterlingData Half Price VCRFlash Until June 14](#)

Until Friday, June 14, 2002 SterlingData's VCRFlash template, which adds VCR navigation buttons to a form, is available for half price. Just put a note in the comments box on the Order Form "less 50% special discount."

Posted Wednesday, June 12, 2002

[etc-III Lodestar Software Discounts](#)

A special order form is now available for all etc-III attendees who would like to take advantage of the discounted pricing for RPM and AFE. This order form will be available through the month of June.

Posted Wednesday, June 12, 2002

[Free MP3 Player](#)

New features in C&G Software's free MP3 player: Button Clear Playlist; Drag&Drop Mp3 files from Windows Explorer to the current playlist; Save/Restore the last used folder (save/open playlist). Code source is now optimized and commented, and will be available shortly for 19\$.

Posted Wednesday, June 12, 2002

[Gitano Software June Drawing](#)

If you purchase gRegPlus during the month of June your name will be entered in a drawing for a chance to win one of 11 prizes. This applies to all gRegPlus purchases, upgrades, competitive upgrades and gRegPlus add-ons. Gitano's competitive upgrade pricing has been reduced giving you a savings of \$100.00 from the regular price. First prize: gSec to complement your gRegPlus purchase. Ten additional prizes will be given of Product Scope 32 PRO. Product Scope 32 PRO provides a combination of tools for keeping track of product, vendor, and customer information that is located on the Internet and on local storage drives. Drawing will be held on July 1st, 2002. All entries must be received by June 30, 2002.

Posted Wednesday, June 12, 2002

[RichReport Gold](#)

Version 1.0 of RichReport is now available. No more bugs/wishes had been reported after the release of beta 4, so the list of changes is quite short. The only new feature is support of sub- and super-scripted text. This update is free for all registered users. The list of supported 3rd

party report addons is still growing. Larry Teames provided template modifications that enable CPCS reports to work together with RichReport. RPM from Lodestar Software works with RichReport without modificatio. The free TinTools previewer has also been tested as well as ClarioNET. The price for the gold release now is US\$99. RichReport is also available from www.clarionshop.com.

Posted Wednesday, June 12, 2002

[Coollook Template Release v2](#)

You can now use Coollook to apply different styles to your applications. You can swap between styles in only one click in the template. All colors, icons, background, button shapes are fully customizable. Coollook release v2.0 is available at the same price as before: US\$49.

Posted Wednesday, June 12, 2002

[Clarion/ASP Online Training Course](#)

If you're looking at purchasing the new Clarion/ASP Templates, than SoftVelocity would like to remind you that the first online training course ("Clarion/ASP in 14 Days") begins next week. The courses are set up on a revolving schedule. You are automatically signed up for the online course when you purchase the templates (only valid during the introductory 30-day offer). After your purchase, you will receive an email from SoftVelocity, outlining the details for the upcoming course.

Posted Wednesday, June 12, 2002

[Free Plugware COM Classes Available Now](#)

The COM Classes written by Andy Ireland of Plugware Solutions are now freely available from ThinkData. Just click on the link for Products and then COM Classes, and you can register to download the classes. An ADO wrapper is included and the supplied example application connects to the MSSQL Northwind database. Just edit your redirection file (C55ee.red) to point to the directory you install the code into and fire up the example.

Posted Wednesday, June 12, 2002

[EasyExcel 1.031 Released](#)

New in EasyExcel 1.031: A bug fix regarding the workbook changing and closing; Changed Init, OpenDoc and Save - added Password and WriteResPassword parameters; New CloseBook and SelectBook methods; Changed Init, OpenDoc and Save code templates - added Password and WriteResPassword parameters.

Posted Monday, June 10, 2002

[XPMenu At ClarionShop](#)

The compXPMenu lookalike is now available from ClarionShop.

Posted Monday, June 10, 2002

[June Simsoft Template Special](#)

From now until 30 June 2002 the Simsoft Templates, Simshape Templates and the Simsoft bundle (of both) will be on special. Pricing will be: Simsoft Templates \$50 (normally \$60) Simshape Templates \$39 (normally \$49) The Bundle \$85 (normally \$95)

Posted Monday, June 10, 2002

[PDF-Tools & Clarionet Reports 2 PDF Demo](#)

A working demo showing how to incorporate report-to-PDF output for ClarioNET, using the PDF-Tools SDK, is now available. Template set is scheduled for release on June 10.

Posted Monday, June 10, 2002

[New "Look Good" Package](#)

Gitano Software has released a "Look Good" package to improve the aesthetics of Clarion applications. Theme Pack 1 includes 12 themes, as well as the gBuddy color template. As a bonus Gitano will design one box shot for your product like the ones on the Gitano site. Also there is a one time 20% discount on any custom graphics work for your application or web site. If you already have the Theme Pack, or IconsXp or gBuddy, then you can get the "Look Good" package at a discounted price.

Posted Monday, June 10, 2002

[Interactive ASP Tutorial](#)

For those wanting some background on ASP, John Armstrong suggests this interactive tutorial.

Posted Monday, June 10, 2002

[Clarion 5.508 \(h\) And ClarioNET 1.2 Available](#)

The Clarion 5.508 (h) patch is primarily a compatibility release to deliver improved support for the ClarioNET 1.2 update. Worth noting is that the enhancement in 5.508 was made to enable ClarioNet to handle correct TAB / z Order on very complex Windows with nested parent structures. The changes to ClarioNet are numerous.

Posted Friday, June 07, 2002

[Clarion/ASP Released](#)

SoftVelocity has released its Clarion/ASP product, which generates Active Server Pages (ASP) web applications with desktop app capabilities. Features include: Complete separation of the User Interface from the Application and Business Logic, allowing easy site integration; Template driven code generation automates ASP development based on your data dictionary; Query by example; Form based data navigation; Form editing procedures with both client and server side data validation; Popup database lookups; User level security; List to List drilldown for data analysis; Scalable to thousands of users

Posted Thursday, June 06, 2002

[SealSoft's xFunction Library v1.5](#)

SealSoft has released xFunction Library v1.5. New in this release: Horoscope functions, including xGetZodiac, xGetChineYear, xSetRusZodiac, xSetEngZodiac, xSetRusChineYear, xSetEngChineYear. By default in xInitLibrary procedure arrays with names of Zodiac Signs and Chinese horoscope are initialized as: Aries, Taurus, Gemini, Cancer, Leo, Virgo, Libra, Scorpio, Sagittarius, Capricorn, Aquarius, Pisces; Ox, Tiger, Cat, Dragon, Snake, Horse, Goat, Monkey, Rooster, Dog, Pig, Rat. You can use xSetEngZodiac and xSetEngChineYear procedures for a changing these names, in the correspondence your native language; New install kits and demos available. Clarion 5/5.5.

Posted Tuesday, June 04, 2002

[DCT2SQL Templates Update](#)

The latest DCT -> SQL templates support two new back ends. DCT2METABASE is a generic ANSI SQL script geared to produce output that can be read into Michael Gorman's metabase product. This is a good starting point to copy and create specific script writing for a backend that is not yet supported. DCR2MSSQL was started because of a private request. This has not been fully tested, and there may be an error in the delete of the relation before the relation create. If you get a chance to test, let Roberto know of any problems you find. WARNING: The Pervasive back end script is still a skeleton. And if you want to make the changes to work correctly please do. You will be given credit, and the variations will go back to the user community for use.

Posted Tuesday, June 04, 2002

[XPMenu Online Purchase](#)

XPMenu is now available for purchase via SWReg, which includes Paypal.

Posted Tuesday, June 04, 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.

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

The RanDum Phrase Generator

Published 2002-06-21

Here's a neat little program by Lee White that generates random phrases, suitable for security applications such as automatically generated passwords. Available options include specifying at least 25% numeric characters, and/or at least 25% mixed case. You can also prevent repeating characters and omit the potentially confusing O,0,I,l, and 1 characters.

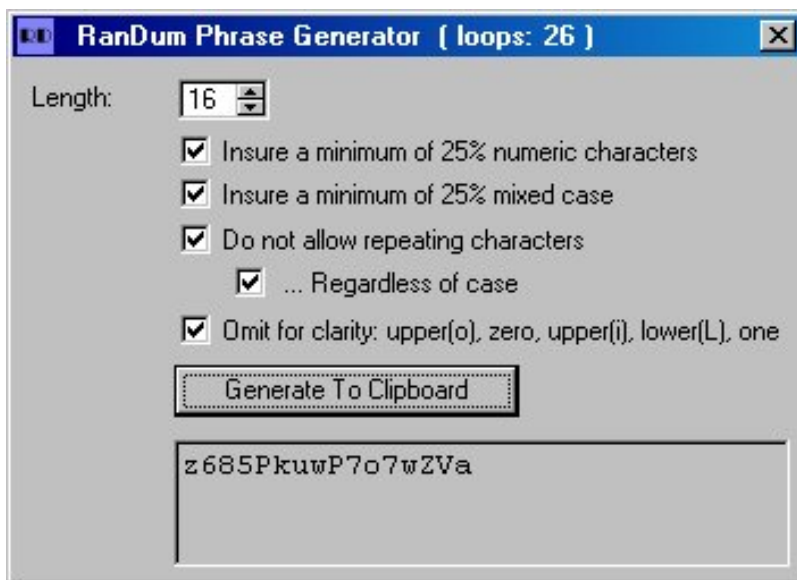


Figure 1. The "RanDum" phrase generator

The RanDum phrase generator takes a brute force approach to creating the phrases. Its starting point is one of the following two strings, depending on whether confusing characters are to be omitted:

```
String57 STRING('abcdefghijklmnopqrstuvwxyza2
                3456789ABCDEFGHIJKLMNPQRSTUVWXYZ')
String62 STRING('abcdefghijklmnopqrstuvwxyza
                0123456789ABCDEFGHIJKLMNPQRSTUVWXYZ')
```

The following loop actually creates the passphrase:

```
LOOP Loops = 1 TO PhraseLen
  Phrase[ Loops ] = CHOOSE(NOT OmitConfusion, |
    String62[ RANDOM(1,62) ], String57[ RANDOM(1,57) ])
END
```

Following this are a series of tests which determine if the phrase meets the specified criteria. If the phrase doesn't pass a test, the generator loop cycles and a new phrase is generated. When a phrase is successfully created, the generator copies it to the clipboard, and displays the number of attempts required in the window title bar.

With the default eight character setting RanDum can generate pass phrases in typically 1-10 tries, which is pretty much instantaneous. If you want to create long phrases, say 99 characters, with all options, RanDum will take a little longer.

```
NCKj2wfwNPN2GA6mw2EPcGLzJxJgwc7a3
N57949fXz2Hqu9UnUVm2m7W6BfuKp4R7Q
64rZNx8oPwMUxzPvYDNtk8wQU9E7VWuw9
```

There are, of course, any number of different ways to approach random phrase generation. If you think you can improve on this code, just post your source as a reader comment below!

[Download the source](#)

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 IV - More Maintaining Order

by **Alison Neal**

Published 2002-06-26

[Last week](#) I outlined how to implement a data structure known as an Ordered Link List and sort it using the very common Bubble Sort. This week I'll begin to cover some of the alternative sort algorithms, including the Cocktail Shaker, Insertion, and Counting sorts.

Cocktail Shaker Sort

The Cocktail Shaker Sort is a variation on the Bubble Sort. It works in the same way as the Bubble Sort, but it works bi-directionally, this means that with every pass at least one element in the list is going to be moved either forward or backward into its correct position.

```

j = pSZ
LOOP i = 1 TO j - 1
  LOOP k = i TO j - 1
    IF pArray[k + 1] < pArray[k]
      swap(pArray[k + 1],pArray[k])
    END
  END
  j -= 1
  LOOP WHILE k > i
    IF pArray[k - 1] > pArray[k]
      swap(pArray[k - 1],pArray[k])
    END
    k -= 1
  END
END
END

```

If I use the same array as the one used for the Bubble Sort, you can see that the result is the same, but the sequence of moves is slightly different:

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

Compare the table above to the table illustrated for the Bubble Sort. Note how the numbers in red (those that have been moved in this step) show up in a zigzag pattern rather than in diagonal lines. This literally shows how the Cocktail Shaker Sort bubbles the highest number upwards (99) and then bubbles the low numbers down (9), rather than re-starting at the beginning every time.

Sort Name	Unsorted Array		Sorted Array	
Bubble Sort	824	Clicks	650	Clicks
Cocktail Shaker	1526	Clicks	1340	Clicks

Unfortunately, depending on the circumstances the Cocktail Shaker Sort can take twice as long as the Bubble Sort.

Insertion Sort

The Insertion Sort or Linear Insertion Sort is quicker than both the Bubble and Cocktail Shaker Sort algorithms, particularly when the data is already in, or is close to sorted order:

Sort Name	Unsorted Array		Sorted Array	
Insertion Sort	613	Clicks	1	Clicks

Note that with 2,500 elements this sort only took 1/100th of a second on my PC to pass through the data. The code for the Insertion Sort is:

```
LOOP i = 2 TO pSZ
    temp = pArray[i]
```



```

j = i - 1
LOOP WHILE j > 0 AND pArray[j] > temp
    pArray[j+1] = pArray[j]
    j -= 1
END
pArray[j+1] = temp
END

```

As you can see, fewer steps are needed to sort the data:

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

The Insertion Sort works by taking the next item and inserting it into the data structure in its proper order with respect to items already inserted.

With the example data, the external loop iterates from 2 to Size (6) times, which is 5 times in total. A copy is taken (`temp`) of the value at the head of the data I'm concerned with (from position 2 to 6). In the first time through the example array, `I` will equal 3, `SZ` will equal 6, and `temp` will therefore equal 99. `J` is made to equal `I - 1`, which is the place previous to the `temp` variable just copied. Then while `J` is not going to make the loop go beyond the bounds of the array and the value in position `J` (22) is higher than the temporary variable, the values will be swapped. In the case of 99, however, this loop will never execute, because 99 is greater than 22 and is therefore already in the correct position in relation to 22.

The next time the main loop iterates, `I` is equal to 3 making `temp` equal 4, the third position in the array. `J` equals 2, which is one less than `I` and is the previous position in the array. Now `J` is greater than 0 and `myArray[j]` equals 99, which is definitely greater than 4. The original `temp` position is made to be 99 and `J` is decremented by one, to equal 1.

At this point the array actually contains 22, 99, 99, 12, 33, 9 and the `temp` variable still contains the 4 for comparison. Back to the nested loop, and 1 is still greater than 0 and `pArray[j]` equals 22 which is also greater than the `temp` value of 4. The value 22 is moved up the ladder, and the array now contains: 22, 22, 99, 12, 33, 9. When the nested loop fails because `J` is no longer greater than zero, the `temp` (4) variable is assigned to the `J (0) + 1` position, and the array is 4,22,99,12,33,9. Thus, with each pass of the loop each element is placed in its correct position in relation to those elements that have already been sorted.

The Insertion Sort Algorithm, like the Bubble Sort and the Cocktail Shaker Sort, suffers time

delays because of the number of comparisons performed. There is one sort that avoids comparisons altogether, but it has other limitations.

Counting Sort

Now compare the results of running this same scenario using the Counting Sort, with 2,500 numbers ranged between 1 – 999:

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

Impressive isn't it? In case you're interested, on my PC, the Counting Sort Algorithm doesn't start to measure clicks until there are 10,000 values to be sorted, and with 50,000 values both the sorted and unsorted arrays only take 12 clicks, that is 12/100th of a second.

The Counting Sort will re-sort the data, going through exactly the same process, regardless of whether the data is already in sort order or not.

```
CLEAR(temp)
LOOP j = 1 TO pSZ
    temp[pArray[j]] += 1
END
LOOP i = 2 TO pMax
    temp[i] += temp[i-1]
END
LOOP j = pSZ TO 1 BY -1
    sorted[temp[pArray[j]]] = pArray[j]
    temp[pArray[j]] -= 1
END
LOOP i = 1 TO pSZ
    pArray[i] = sorted[i]
END
```

The `temp` array is declared to the size of the highest number in the range used. This is the main limitation of this algorithm. It has a heavy memory overhead if you're handling exceptionally large numbers, but for numbers between 1 and 999 it works fine, in my experience.

Take an array with the values 4,4,2,3,2. It has a possible maximum number of 5 and a size of 5. After the first loop the `temp` array looks like this: 0,2,1,2,0. Note that this is fundamentally

a count, there are no 1s in the array, and so the value of position one is zero. There are two twos in the array so the value of position two is 2. There is one three, there are two fours and no fives.

The second loop provides the positioning for the values for when the array is sorted. At the end of the loop `temp` would look like this: 0,2,3,5,5. This basically tells me that the last position of the two twos is position two; therefore they will fill spaces one and two in the sorted array, the last position of the 3 is position three, and the last position of the fours will be position 5. The algorithm will never read up as far as position 5 in the `temp` array so it can be ignored, and even if there were a 5 then the 5 previous slots filled plus 1 (the count value of the fives) equals 6, making the last position for the value 5 the sixth space in the sorted array.

As the array is sorted into order in the third loop the position counts are decremented, so in the case of the two 2 values, the algorithm puts the values in their appropriate places.

When `J` equals 5, `pArray[5] = 2`, and `temp[2] = 2`, so `Sorted[2]` is made to equal the value 2. Then `temp[pArray[5]]`, which is `temp[2]`, is decremented by 1, so that when `J` equals 3, `pArray[3] = 2`, `temp[2] = 1`. `Sorted[1]` is also made to equal 2. In the end the `temp` array looks like 0,0,2,3,5 and the `Sorted` array looks correct with 2,2,3,4,4. However I needed the original array so that I could calculate which value went where. I now have to copy the sorted array over our original array to finish in the final loop.

What happens if some of the values to be sorted are zeros? If this is the case then you will need to add/subtract one to the array indexes where appropriate.

Summary

As usual there are several different ways to skin the same cat. Some however are far more efficient than others, and some are going to be quicker under different circumstances.

It will always be more efficient to maintain a list in order from the start, rather than having to sort and re-sort it. Unfortunately, I don't always have that luxury.

In my next article I will be continuing with the sorting algorithms, and showing you my personal favourite: the Shell 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](#)

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**tm

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

[Topics](#) > [Internet](#) > [Email](#)

Emailing PDF Files From A Report Previewer

by George Lehmann

Published 2002-06-28

In recent months, a number of third-party products have come into the marketplace that allow the user to save a report in a standardized format, and then email that report to someone. Unfortunately, from my point of view, every one of these products suffers from one common shortcoming: Lack of integration with the report previewer.

Before the hoots and objections start flying about how this product and that product is compatible with various report previewers, let me share what I think to be the expectation of a typical user, if there is such a creature.

First, the user has a need to answer some question about his or her business, and you have written a report to provide that answer. The report will almost always have some options, such as a date range or an include/exclude switch, such that the user may need to run the report more than once before getting the desired results. The user may even need to leave the report and do some data entry work to clean up the reported data before the report is in a suitable state for delivery to a third party. Then there is that comfort factor of seeing the preview one last time before the report is printed (or now emailed). On top of all that, the report might require an appreciable amount of time to generate.

Given this scenario, forcing the user to make a decision regarding saving or sending the results before the report is prepared and reviewed would likely be perceived as overly restrictive. When users are so sensitive that they complain about a few unnecessary (at least to them) keystrokes just to complete a form, I can only imagine the complaints I'd receive if I made my users run reports twice every time they wanted to email something *and* be confident the results were what they wanted.

What I really want is to have an Email button on the Previewer screen. After the users run a

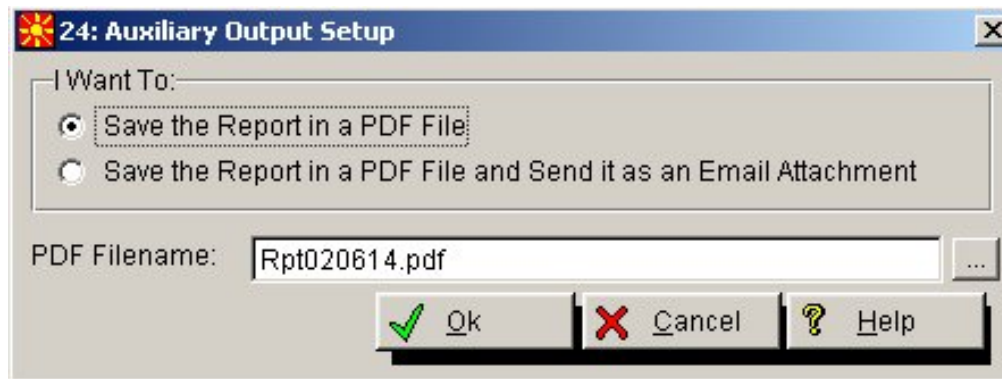


Figure 2. The Auxiliary Output Setup form

After studying the code generated by the PDF-Tools report extension template, I copied the necessary snippets into a source procedure aptly named `Aux_Output_PDF`. The central concept in the whole process is to take the preview queue, which is really just a queue of filenames that point to a number of separate WMF files containing the report image, and transform those WMF files into the desired output format. After a few tweaks to the PDF-Tools source (which John has now incorporated into his Version 2.0 product), I was able to call the PDF-Tools object methods, with the preview queue and the paper orientation from the RPM previewer, to generate PDF files. This all culminates in a very short source procedure to create the actual PDF file as follows:

```
Aux_Output_PDF  PROCEDURE  (String FileName, |
                        Queue PreviewQ, Byte IsLandscape)
PDFX            PDFXToolsReportClass
CODE
PDFX.Init(FileName)
PDFX.SetPaper(0, 0, 0, 0, IsLandscape)
PDFX.SetPreview(0)
PDFX.Generate(PreviewQ)
```

After this procedure completes, I've got a PDF file, and I know its name. I can now create a simple email window (as per NetTalk instructions), and fill in my PDF filename as the attachment. The user can then fill in his who-to address and some short comments, and send the email. I store a few values globally which the NetTalk template otherwise asks the user to enter, so my email window wound up looking like Figure 3.

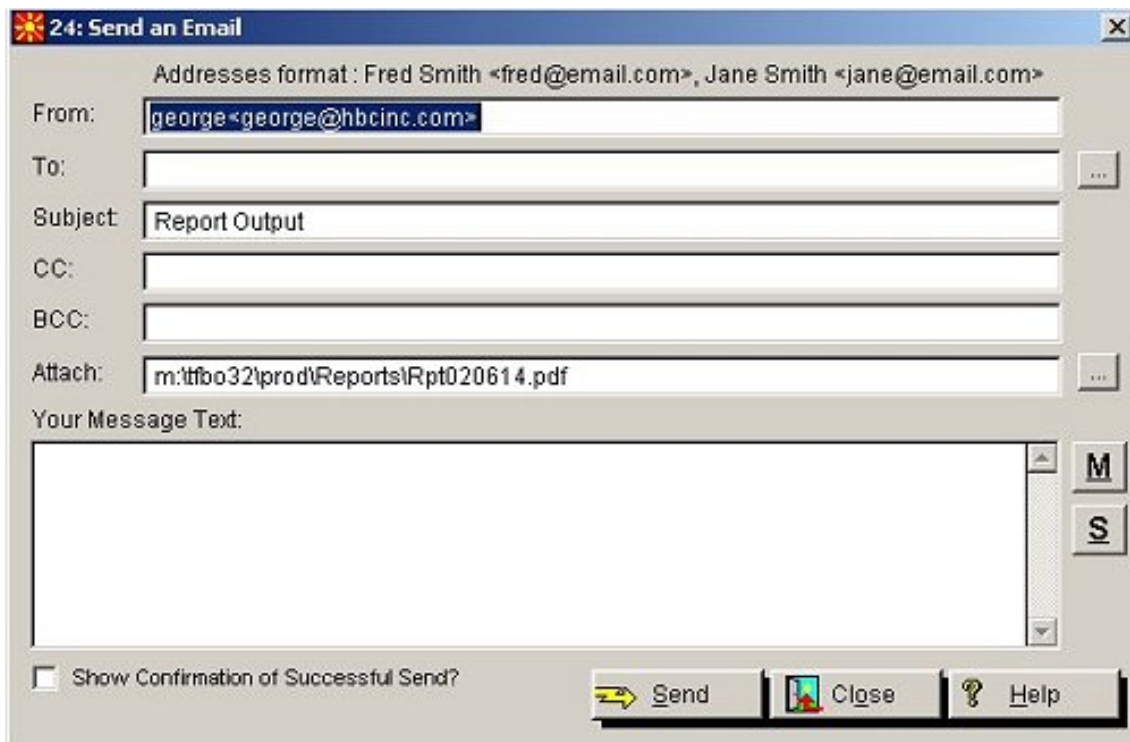


Figure 5. An email window using NetTalk

The only piece of the puzzle left was now to connect the RPM developer button to my `Aux_Output` procedure. Because the RPM developer button is nothing more than a hook to call my code, I had to alter the RPM DLL to make this connection. Fortunately, Lee White gives good instructions on editing his RPM5TA2B project to accomplish this. After copying the RPM source files into a work directory, I inserted code into the otherwise empty `RPM_Developer` procedure within the `RPM-5t-abc-b.CLW` source file to call my `Aux_Output` procedure, as follows:

```
RPM_Developer PROCEDURE(Queue PrintPreviewQueue, |
                        LONG StartPageNumber, LONG EndPageNumber, |
                        BYTE PagesOnScreen, |
                        BYTE PageSelected, SHORT MenuChoice)
CODE
Aux_Output(PrintPreviewQueue, InternalGroup.RPT_Landscape)
RETURN
```

Note that the only code I added was the one line calling `Aux_Output`. The rest of what is shown is the shell of the procedure provided by RPM.

Next, I added a prototype for `Aux_Output` in this same source file, in the MAP section, identifying the `.LIB` file containing `Aux_Output`:

```
MODULE('utilops.lib')
Aux_Output(Queue, Byte), DLL(dll_mode)
END
```

To finish turning this into an executable binary, I added the same .LIB file to the project files list and compiled the RPM5TA2B project, producing a new DLL with the call to my Aux_Output procedure. Not wanting to overwrite the original RPM5TA2B.DLL file, I copied this new DLL into my development and distribution directories where it would be picked up ahead of the original DLL, leaving the original DLL untouched in the \C55\BIN directory.

The next step was to modify the default icon in the RPM5TA2A.DLL file (yes, this is a different DLL from what was modified in the last step) using an icon editor. This is strictly a cosmetic step, replacing the default exclamation mark icon provided in RPM with something that seemed more appropriate for emailing a report. I chose a 3-way blue arrow design as shown below on the fourth button from the right, hoping to achieve some symbolism about “alternate output directions.”



Figure 4. The modified RPM toolbar

The last step was to change the template values in the RPM extension template in my application to enable the Developer button, which is otherwise hidden. This did require opening every report in my application and changing the settings on the RPM template, as well as changing the defaults for next use on the global RPM template, but after I got the “click-rhythm” down, each report took only about five seconds to modify.

Now that little three-way arrow is a pathway leading from my application directly onto the email ramp of the information superhighway. And most importantly, I have a universally consistent report tool for the end users, which is available to them at the exact moment when they feel the need to reach for it.

George Lehmann is Vice President of [Horizon Business Concepts, Inc.](#), which he co-founded in 1988 to develop PC-based software for aviation businesses. George has a Master's Degree in Computer Science from the University of Illinois at Urbana-Champaign, and has been programming in Clarion since 1998. For relaxation he says nothing can beat an afternoon on the highway with his wife Leann and their Harley Road King.

Reader Comments

[Add a comment](#)

Addendum: This entire process works equally well with...

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](#) > [Databases](#) > [MySQL](#)

Clarion And The Fatal MyODBC/MySQL Flaw

Published 2002-06-28

I've been a satisfied user of MySQL (and holder of a paid license) for several years now, and I've written [several articles](#) on how to set up and use this popular open source database with Clarion. Recent events, however, have caused me to reconsider my choice of open source database. I think it's still possible to use MySQL effectively with Clarion, but only under certain conditions.

MySQL is, at present, the database behind the web application that delivers Clarion Magazine. The database contains, at present some 50 tables for everything from article and subscriber information to email templates and topical index entries. And MySQL has been an excellent choice for the web application itself, which uses the [mmMySQL](#) JDBC driver.

Of course, an online magazine about Clarion that doesn't require a lick of Clarion code would be heresy. There's another vital application, written in Clarion, with which I administer the magazine database. I use that application to add articles, set rights, override/update subscriber data, and much more. That application uses the MyODBC driver.

The problem

The Clarion Magazine web server, a Linux box, resides in my downtown Winnipeg office, and until a few months ago that meant it was just a few feet away from my Windows development machine. Then I began working more often from my home office, about a ten minute drive away. Consequently I had to connect to the MySQL database over the internet, and although I have a cable modem at home, that was still a lot less bandwidth than the 100Mb I was accustomed to. Browsers that had been pretty snappy at the office were now dog slow. For instance, one browse with 600 records, averaging 400 bytes each including a text field, was taking up to 8 seconds to load.

I wasn't the first to notice this problem with MySQL. Others had reported performance issues

with really large tables over LAN connections. I only ran into trouble once my network bandwidth went from 100Mb to about 500Kb, because my tables weren't that huge.

The driver

I use the [MyODBC](#) driver with Clarion because that's the only way I know of accessing MySQL data files with ABC. But as it turns out, MyODBC has a fatal flaw that makes it unusable with really large data sets.

First, a little background on Clarion's standard approach to SQL and ODBC (and my thanks to Scott Ferrett for setting me straight on a few points). Say you have a table with five hundred thousand records. If this is a TPS file, and you open an unfiltered browse with, say, 20 lines of data, the file driver will simply retrieve the first 20 records from the table. If you do the same with an SQL table, the file driver (most likely the ODBC driver) will send a `SELECT` statement to the SQL server. The server will locate all the possible records and store this information in a result set or record set. The driver will then, typically, ask for the first 20 records from that result set (or it may retrieve a few "pages" at a time).

Ideally, MyODBC would exhibit this same behavior, but instead, when you open that hypothetical browse, and the Clarion ODBC driver issues the `SELECT` to the MyODBC driver, the internal MyODBC function that MyODBC calls is `mysql_store_result`. This function retrieves all the data in the result set from the MySQL server. So if the `SELECT` statement essentially asks for all of the data, all 500,000 records will have to shuttle across your network connection before MyODBC can return the first 20 to your browse.

When I first saw the source code that called this function I was astounded. I thought there had to be a way of only retrieving the first page of data instead of everything in the result set. And in fact there is a function called `mysql_use_result` which is intended to do just this.

The only difficulty with `mysql_use_result` is it won't work with ABC, at least not at present. In fact, the option to use it is now commented out in the driver. And don't bother uncommenting the source. I'll explain why in a moment.

Ultimately, I think, MyODBC is unsuitable for large tables in Clarion because MyODBC's authors decided to build it on top of the MySQL client library, which contains the code for `mysql_use_result` and `mysql_store_result`. This client library also has only one buffer for the currently active result set. That's sufficient if you're building a command line query tool, because you only use data from one query at a time. But a Clarion application may want to have data from several queries in memory at any one time.

Consider a browse procedure with parent and child browses. In a normal ODBC/SQL setting, when you open the first browse, the driver will issue a `SELECT` to the back end, get a result set

(most of which stays on the server), and retrieve records back from that result set as needed. The driver does not need to keep issuing new `SELECT` statements to the back end. Each time the parent browse's selection changes, of course, the driver will need to issue a new `SELECT` for the child browse, so that the related data can be displayed. The parent's result set stays open while the driver requests a new result set for the child browse each time the parent selection changes.

Here's the kicker. If you enable the use of `mysql_use_result` in this scenario, you'll get a "command out of sequence" error as soon as the second `SELECT` executes. This is because there is only one active result set buffer in the MySQL client library, and the library code detects that there are still records in that buffer from the first `SELECT`. The only way to avoid the error is to either retrieve all of the data in the result set, or cancel the select. In either case you no longer have an open result set on the first browse, which was the whole point.

Consider the following [excerpt](#) from the MySQL docs:

An advantage of `mysql_use_result()` is that the client requires less memory for the result set because it maintains only one row at a time (and because there is less allocation overhead, `mysql_use_result()` can be faster). Disadvantages are that ... you must retrieve all the rows even if you determine in mid-retrieval that you've found the information you were looking for.

I believe canceling the query is also an option. Here's another [quote](#):

If you use `mysql_use_result`, you have to ensure that no other thread is using the same connection until the result set is closed. However, it really is best for threaded clients that share the same connection to use `mysql_store_result()`. If you want to use multiple threads on the same connection, you must

Why Open Source?

I've been asked why I don't use a mainline commercial database such as MS SQL instead of MySQL. When I began Clarion Magazine, one of the first choices I made was the platform. For various reasons, primarily performance and reliability, I settled on Linux. And although there are a number of database choices for Linux now, that wasn't always the case. At first, the only practical alternatives were MySQL and PostgreSQL, and when I began using Linux, PostgreSQL had an 8k string size limitation (long since remedied) which made it impractical for my purposes. That left MySQL.

There are both commercial and non-commercial database options for Linux now. If anything, however, this experience has made me more convinced that open source is a good choice. Faced with a performance problem in MyODBC, I was able to look at the source for both MyODBC and MySQL to determine what was happening. I also built and tested several MyODBC

have a mutex lock around your `mysql_query()` and `mysql_store_result()` call combination. Once `mysql_store_result()` is ready, the lock can be released and other threads may query the same connection.

Alarming, the MySQL manual also says the [following](#):

On the other hand, you shouldn't use `mysql_use_result()` if you are doing a lot of processing for each row on the client side, or if the output is sent to a screen on which the user may type a ^S (stop scroll). This will tie up the server and prevent other threads from updating any tables from which the data is being fetched.

You may not use `mysql_data_seek()`, `mysql_row_seek()`, `mysql_row_tell()`, `mysql_num_rows()`, or `mysql_affected_rows()` with a result returned from `mysql_use_result()`, *nor may you issue other queries until the `mysql_use_result()` has finished.* (emphasis added)

Clearly `mysql_use_result` is a can of worms best left unopened. And that leaves `mysql_store_result`.

What are the alternatives? Is there any way to improve performance by restricting the result set on the server?

What about LIMIT?

MySQL `SELECT` statements can have a [LIMIT](#) attribute which restricts the data returned by the server.

The `LIMIT` clause can be used to constrain the number of rows returned by the `SELECT` statement. `LIMIT` takes one or two numeric arguments. The arguments must be integer constants. If two arguments are given, the first specifies the offset of the first row to return, the second specifies the maximum number of rows to return. The offset of the initial row is 0 (not 1):

So yes, you can use `LIMIT`, but what you need to keep in mind is that you're restricting the data in the result set, so from your application's perspective, if you add a `LIMIT 50` to your `SELECT` (one way is via the `SQLCallbackInterface` - see `SQLCallback` in the Clarion Help for details) all that happens is the rest of your data magically disappears, as far as the

driver modifications.

Besides, MySQL and PostgreSQL, the only major open source database on Linux at present would seem to be Interbase, and it isn't widely used in the web development world. Also there seems to be some question about the future of the open source version, but I'll keep my eyes open.

browse is concerned. You can page down all you want but you won't see more than those 50 records. Any effective use of `LIMIT` will have to be a bit more sophisticated than that.

Kristian Hyllestad has a [demo of the legacy templates](#), using `LIMIT`, so evidently this can be done. I haven't tested the code, and don't know if anyone has made, or will make, the template changes required to support this in ABC.

Other approaches

If you don't have a lot of code already using MyODBC, or you're willing to rewrite your procedures, have a look at Dan Presnell's [twMySQL](#). This product allows Clarion apps to connect directly to MySQL servers without using MyODBC. The templates are ABC/Legacy compatible.

The ideal solution, in my opinion, is a rewrite of the MyODBC driver which does not rely on the MySQL client library, or a rewrite of both MyODBC and the client library to support concurrent result sets. The [mmMySQL](#) JDBC driver is open source – perhaps someone could port it to ODBC. But I'm not holding my breath.

Conclusions

I've relied heavily on MySQL for several years now, and in general I've been quite happy with the results. But now that I need to work with my databases remotely, and I want to continue to use my ABC-based code, I find MyODBC's lack of concurrent result sets unacceptable. I've posted private and public messages to the developers and have not received any replies or other information indicating a update is in the works, so as far as I know the prospects are not good.

If you have large data sets or a slow network connection, and you must use ABC, I cannot recommend the combination of MySQL and MyODBC at present. As I have the latter condition to deal with when running the ClarionMag admin program, I'm looking at other options for my Linux server, in particular [PostgreSQL](#). I've briefly tested the [psqlODBC](#) driver, and it does work the way I wish the MyODBC driver worked. I expect I'll have more to write about PostgreSQL in the coming months.

Update!

Following some discussions in comp.lang.clarion with Steven Spierenburg, I took another shot at a browse class that uses LIMIT. I don't hold out huge hope for this, but perhaps someone with a better understanding of ABC internals can take this on and make something workable out of it.

The cciMySQLFileManager class is derived from FileManager, and simply implements the SQLCallback interface to append a LIMIT 50 on the select, i.e.

```
return instr & ' limit 50'
```

For testing you may want to change this to some other value. I don't know that it's ideal to keep it to the exact number of records in the browse - I suspect you'd want to keep a couple of pages cached.

The other class is cciMySQLBrowseClass, derived from BrowseClass. My strategy was to mimic the behavior you get when coming back from an update - in that case, the browse is reloaded from disk. But it's not working quite right, probably because the code is being called when no more records can be found.

Also, and most importantly, it's not immediately obvious to me how to implement a page or scroll up.

I'll be more than happy to test any improvements, but unfortunately I don't have much more time to devote to coding a workaround right now. These classes will get me over the hump, and because I'm running a Linux database server I expect to port to PostgreSQL, pending further tests (no doubt a solution will eventually emerge, one way or another - I just can't wait that long).

To use these classes, just change the File Manager for the MySQL table in question to cciMySQLFileManager, and the Browse Class for the browse to cciMySQLBrowseClass. You may need to refresh the ABC class list.

[Download the classes](#)

Reader Comments

[Add a comment](#)

I appreciated your clear and oportune post. Sadly to say,...

A few further notes on the performance differences I've...

Hi Dave, Thanks for this article. I have just received...

Rex, I did all my testing with MyOCBC 3.51.03, which at...

I've posted derived browse and filemanager classes (see...

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.