

[Home](#)   [COL Archives](#)

[Advanced Search](#)

[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)

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

[Advertise](#)  
[Reader Comments](#)  
[Call for Articles](#)  
[Log In](#)  
[Membership/Subscriptions](#)

[FAQ](#)  
[Privacy Policy](#)

[Contact Us](#)  
[PDFs](#)  
[Freebies](#)  
[Open Source](#)

## Clarion 5.507 Service Release 7 (Final candidate)

The Final Release candidate for Clarion 5.507 is now available for download. This build resolves a problem in Microsoft Windows XP compatibility and includes several additional fixes since the Jan 9 Release. This release can be downloaded by all customers, but SoftVelocity is naming it a candidate release while 3rd party developers update their add-on tools as needed. SV expect to make it official on Tuesday of next week. Note for Enterprise users, this release also includes an update to Data Modeller with fixes and enhancements

*Posted Friday, March 01, 2002*

## SV To Announce Upcoming Products

Next week SoftVelocity will be releasing some announcements on upcoming product releases, training classes, workshops and other items of interest.

*Posted Friday, March 01, 2002*

## ClarionMag Sweepstakes Final Prizes Awarded

The Grand Prize Winner in the Clarion Magazine Sweepstakes is Nardus Swaneveler, who has elected to take an [ETC III](#) conference registration as his prize. Congratulations, Nardus! Second and third prizes have also been awarded.

*Posted Monday, March 04, 2002*

## Using Real Icons In The Listbox

### News

[twTools 1.1](#)

[Gitano Build G Updates](#)

[CapFlash v3 Released](#)

[INN Bio for 14-Mar-2002](#)

[RPM C5.5.07 Update Available](#)

[EasyVersion 1.01 Released](#)

[EmailReport 2.1](#)

[xAppWallpaper Manager v1.5](#)

[Buggy 3.1 Available](#)

[twTools For Clarion](#)

[ConVic 2002 March 23-24](#)

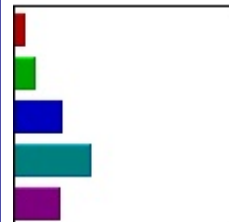
[New Generation Of SetupBuilder Install Engines Test](#)

[Clarion Skill Pool Web Site](#)

[Clarion 5.507 Released](#)

### SURVEY

How important is .NET to your current or near future needs?



■ Vital (6%)  
 ■ Very important (10%)  
 ■ Somewhat important (23%)  
 ■ Unimportant (36%)  
 ■ What is .NET? (22%)

286 Total Votes

[Previous Surveys](#)



### **Header - Part 3**

For quite some time Steffen Rasmussen has been working with the idea of sorting listboxes by clicking on the header, and showing the sort order with an icon. His previous solution (described in Clarion Magazine) used characters in the header to show sort orders; in this series, Steffen shows how to use real icons via buttons. Part 3 of 3.

*Posted Tuesday, March 05, 2002*

### **Weekly PDF for February 24 - March 2, 2002**

All ClarionMag articles for February 24 - March 2, 2002 in PDF format.

*Posted Tuesday, March 05, 2002*

### **Migrating The Inventory Application To SQL Server (Part 4)**

In the previous fourth installment in this series, Ayo Ogundahunsi converts the States data for the Inventory application from a flat file model to a SQL relational model.

*Posted Thursday, March 07, 2002*

### **Clarion Technology Previews And Workshop April 20-22**

SoftVelocity will hold a Clarion Technology Workshop and Training Seminar in Ft Lauderdale on April 20-22, 2002. This is the first in a series of limited availability developer events SoftVelocity expects to host regionally (U.S.) and internationally in 2002. In small gatherings that above all emphasize training, You'll receive a very thorough introduction to the new Clarion technologies. Standardized training will also be available. Technologies to be covered include ADO support, SQL access templates, and XML support, plus the Clarion/ASP and the Clarion/PHP web scripting products. New Clarion 5.6 functionality will also be presented.

*Posted Friday, March 08, 2002*

[Free .NET Development Environment](#)

[Accounting ComCode Version 5.1 Released](#)

[Taboga RTFWriter Library](#)

[Taboga Barcode Library 1.1](#)

[twTools for Clarion Pre-Release Demo](#)

[INN Bio for 6-Mar-2002](#)

[PDF-XChange & Tools Last Chance Pricing](#)

[BigTamer\(tm\) Templates And Utilities](#)

[GUI.IT Template Beta Demo](#)

[SysCalPack Bundle Available](#)

[New Bundles From Gitano Software](#)

[Legacy Apps And MySQL](#)

[Linder SetupBuilder 4.01 Receives Softmag's Editor's Choice](#)

[CPCS Releases Excel/ASCII Output Addon](#)

### **etc-III - 10 Weeks And Counting...**

etc-III is now just 10 weeks away. Features include: education from acknowledged leaders in our community; tranquility of the Great Smoky Mountains National Park; camaraderie with other developers and friends. SoftVelocity's Bob Zaunere is the keynote speaker.

*Posted Monday, March 11, 2002*

### **Weekly PDF For March 3-9, 2002**

All ClarionMag articles for March 3-9, 2002 in PDF format.

*Posted Monday, March 11, 2002*

### **The Previous Surveys Page Returns**

The Previous Surveys page is back! Just look for the link at the bottom of the survey on the main page (or click above). This page contains links to all previous survey results, and is always current.

*Posted Tuesday, March 12, 2002*

### **The Program's Finished. Now What?**

Okay, you've finished the greatest piece of software since, uh, Clarion? It's a program that millions of users will want to install on their systems, and at only \$19.95, have you got a deal for them. There's only one drawback: How will they learn about this fabulous product? Marketing, that's how. But who will do it? Braxton DeGarmo lays out the options.

*Posted Wednesday, March 13, 2002*

### **Creating A SCADA Interface With Clarion (Part 1)**

There are many industries that rely on automated process controls for creating a product or providing a service. Supervisory Control and Data Acquisition (SCADA) is the term often used to describe an industrial automation system. Sometimes all of that acquired data needs to be stored somewhere for future reference; since

[Linder SetupBuilder](#)

[4.02 Web Edition](#)

[Beta 1](#)

[DOS Printer Version](#)

[6 Released](#)

[Gitano Requires Paid](#)

[Email Accounts For](#)

[Online Purchase](#)

[EasyDocker 1.0](#)

[gCal 3.0 Update Now](#)

[Available](#)

[xNotes v1.3](#)

[Released](#)

[xTipOfDay v1.4](#)

[Released](#)

Clarion is very good at storing, manipulating, and viewing data, a Clarion application is a good fit for a SCADA system data storage interface. Part 1 of 2.

*Posted Thursday, March 14, 2002*

### **Secondary Forms (Part 1)**

Sometimes there are too many fields to place on a single window. Sometimes fields can be grouped according to a logical division of data types. Sometimes, spreading fields so that not all of them are displayed simultaneously makes the data easier to digest. Sometimes the user wants multiple screens and sometimes it just looks better to use multiple pages to display data.

*Posted Friday, March 15, 2002*

### **Weekly PDF For March 10-16, 2002**

All ClarionMag articles for March 10-16, 2002 in PDF format.

*Posted Tuesday, March 19, 2002*

### **Secondary Forms (Part 2)**

Sometimes there are too many fields to place on a single window. Sometimes fields can be grouped according to a logical division of data types. Sometimes, spreading fields so that not all of them are displayed simultaneously makes the data easier to digest. Sometimes the user wants multiple screens and sometimes it just looks better to use multiple pages to display data.

*Posted Wednesday, March 20, 2002*

### **Creating A SCADA Interface With Clarion (Part 2)**

There are many industries that rely on automated process controls for creating a product or providing a service. Supervisory Control and Data Acquisition (SCADA) is the term often used to describe an industrial automation system. Sometimes all of that acquired data needs to be stored

somewhere for future reference; since Clarion is very good at storing, manipulating, and viewing data, a Clarion application is a good fit for a SCADA system data storage interface. Part 2 of 2.

*Posted Thursday, March 21, 2002*

### **Baby Overrides Publishing Schedule!**

The next issue of Clarion Magazine will appear the week of April 8, 2002. Clarion Magazine's publisher, Dave Harms, is taking some time off to be with his wife Bonny and their newborn daughter Sophia Elizabeth Nicole.

*Posted Friday, March 22, 2002*

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.

[Home](#) [COL Archives](#)[Advanced Search](#)[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)[Related Categories](#)[Home](#)[Topics](#)[Related articles](#)[Back Issues](#)[COL Archives](#)

## ClarionMag Sweepstakes Final Prizes Awarded

[Advertise](#)Published 2002-03-04 [Printer-friendly version](#)[Reader](#)[Comments](#)

The Grand Prize Winner in the Clarion Magazine Sweepstakes is **Nardus Swanevelder**, who has elected to take an [ETC III](#) conference registration as his prize. Congratulations, Nardus!

[Call for Articles](#)[Log In](#)

Second prize of a two year subscription/renewal goes to **Roxanne Griffin**.

[Membership/ Subscriptions](#)[FAQ](#)

Third prize of a one year subscription/renewal goes to **Peter Gibson**.

[Privacy](#)[Policy](#)

The following people are winners in the Clarion Magazine Sweepstakes interim draw:

[Contact Us](#)[PDFs](#)

- Six month Clarion Magazine subscription/renewal: **Eric Griset**

[Freebies](#)

- Six month Clarion Magazine subscription/renewal: **David LeYanna**

[Open Source](#)

- Six month Clarion Magazine subscription/renewal: **Gary Stanley**

- Six month Clarion Magazine subscription/renewal: **Uro Mencinger**

- Six month Clarion Magazine subscription/renewal: **Nick Tsigouro**

- [CW Assistant](#) utility, valued at \$99: **Ramon Reed**

- [Clarion Source Search](#) utility, valued at \$45: **S. Hills**



- [Clarion Source Search](#) utility, valued at \$45: **Janice Cournoyer**
- [G-Cal](#), valued at \$99: **Chantal St. Jean**
- [G-Calc](#), valued at \$69: **Sherae Gronbach**
- [G-Buddy](#), valued at \$99: **Antonio Oliveira**

Congratulations to all the winners in the Clarion Magazine Sweepstakes!

## 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.

[Home](#) [COL Archives](#)
[Advanced Search](#)
[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)
[Home](#)
[Back Issues](#)
[COL Archives](#)
[Advertise](#)
[Reader](#)
[Comments](#)
[Call for](#)
[Articles](#)
[Log In](#)
[Membership/ Subscriptions](#)
[FAQ](#)
[Privacy](#)
[Policy](#)
[Contact Us](#)
[PDFs](#)
[Freebies](#)
[Open Source](#)
[Topics](#) > [Browses](#) > [Browses, Using](#)

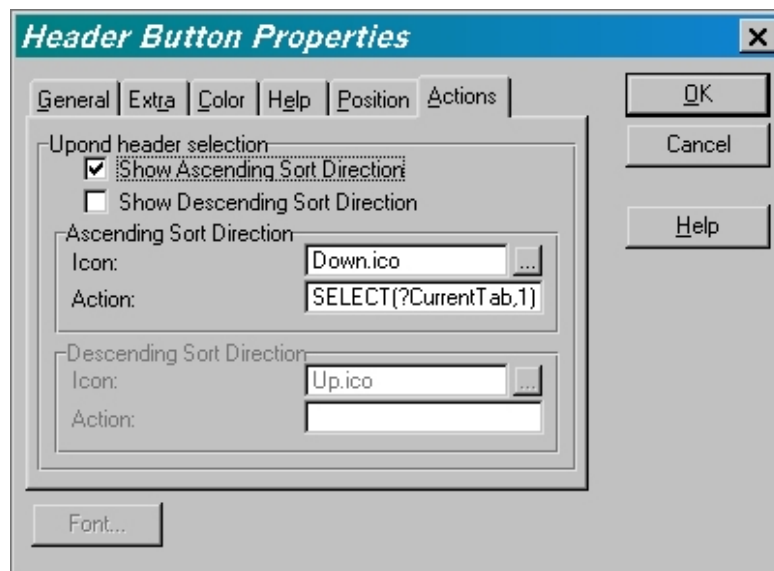
## Using Real Icons In The Listbox Header - Part 3

 by **Steffen Rasmussen**

 Published 2002-03-05 [Printer-friendly version](#)

In Part 1 of this series I began my quest for real icons in list box header columns with a lot of source code that showed how to achieve this using buttons in place of header columns. In Part 2 I showed how to create a template to do most of the work. There's one more task to finish, and that's associating source code with a dynamically created control.

The action tab in the template is the part that distinguishes this template from the normal button template.



**Figure 6. Actions tab**

Here you will find two primary template options in the form of check boxes. One for selecting ascending and one for selecting descending sort direction. In other words here you have the option to select:

### Related Categories

[Browses, Using\(28\)](#)
[Filtering/Ordering\(27\)](#)
[Filtering/Ordering\(27\)](#)
[User Interface\(10\)](#)

### Related articles

[How ABC Handles Multiple Sort Orders](#)
[How ABC Handles Multiple Sort Orders \(Part II\)](#)
[How ABC Handles Multiple Sort Orders \(Part III\)](#)
[Alias - Who Was That Masked File?](#)
[ABC Design Series: The ViewManager Part 1](#)
[The Clarion Advisor - Listbox Styles](#)
[ABC Design Series: View Manager Part 2](#)
[How To Handle Additional Sort Orders](#)
[List Box Marking](#)
[The Clarion Advisor: Adding/Removing Key Tabs](#)
[The Clarion Advisor: Browse Popup Menu Tricks](#)




- Not to show sort direction
- Show only ascending sort direction
- Show only descending sort direction
- Show both ascending and descending sort direction

When you choose to show a sort direction your will also have to select an icon to represent the sort direction. I will elaborate this in a short while. For the time let's stay with the two options.

The options you select in the template must also be reflected in the generated code for the procedure. You can use the template #IF statement to accomplish this:

```
#IF (Check box for ascending sort direction has been selected)
  #IF (Check box for descending sort direction has been selected)
    Use code for both conditions
  #ELSE
    Use only code for ascending condition
  #END
#ELSE
  #IF(Check box for descending sort direction has been selected)
    Use only code for descending condition
  #ELSE
    Use no code
  #END
#END
```

Where do you then place this code? Under normal circumstances there is always attached an embedded point for any given control, but in this case the control does not exist until the time of execution of the program. Instead you always have the possibility to look for an Accepted event and in this case, the event is the field equate of the dynamically created control:

```
#AT( %WindowManagerMethodCodeSection, 'TakeEvent', '(),BYTE')
CASE EVENT()
OF EVENT:Accepted
  CASE ACCEPTED()
  #SET(%ColumnNumber, 0)
  #FOR(%ControlField)
    #SET(%ColumnNumber, %ColumnNumber+1)
  OF LOC:HeaderButton%ColumnNumber
```

And, of course, you have to be able to respond to events for the created buttons. This code has to perform the following:

- Set icon
- Perform action when selected
- Remove Icon from previous selected button

For example, the most complicated operation is the one where you want two different sort directions for the same button. In

[Sidebar Menus](#)

[Creating An MS Outlook-Style Menu In Clarion](#)

[Creating An MS Outlook-Style Menu In Clarion: Part 2](#)

[Outlook Menu Templates](#)

[The Clarion Advisor: Copying Browse Boxes Between Procedures](#)

[Conditional Sort Orders and Page Breaks in Reports: Part 1](#)

[The Clarion Advisor: Displaying List Totals](#)

[Conditional Sort Orders and Page Breaks in Reports: Part 2](#)

[Converting PowerBrowse from Legacy to ABC](#)

[Completely Dynamic Report Orders and Breaks Part 1](#)

[Completely Dynamic Report Orders and Breaks Part 2](#)

[Look Ma, No Keys!](#)

[Windows-Style List Box Sorting](#)

[The Clarion Advisor: Locating Records With PROP:SQLFilter](#)

[Dynamic Filters: The Theory Behind the Facts](#)

[Dynamic Filters:](#)

pseudocode it would look like this:

```

IF Previous selected button = selected button
  IF the Button has been selected before
    Button{PROP:Icon}='Descending Icon'
    Action for descending selection
  ELSE
    Button{PROP:Icon}='Ascending Icon'
    Action for ascending selection
  END
ELSE
  Remove Icon from previous selected Button
  Button{PROP:Icon}='Ascending Icon'
  Action for ascending selection
END
Set Current Button = Previous Button

```

Although I have used the terms Ascending and Descending for the sort direction, it is up to you to define what the action actually is. As you can see in Figure 6 the template allows you to enter some Clarion code for the action. For example, when you have selected a button for the first time the action you have defined for the ascending order could be to select a tab in a sheet:

```
SELECT(?CurrentSheet,2)
```

Now if you have used the Clarion wizard to make your application, a sort condition will already have been created for the particular tab. So when you make the header button select this tab, the list box is sorted accordingly. In this way the template does not interfere with the existing templates, wizards or wizatrns. On the other hand, if you want a sort action that is not already represented by a tab, you may find it easiest to just set up a tab and a sort order in the browse's actions, and then select the tab from the button template. The actual browse tabs are not necessary any more for the user so you can just hide them by selecting the wizard check box in the sheet properties.

[Applying The Theory](#)

[SetFilter to the Max](#)

[Creating Elliptical Windows in Clarion](#)

[Tip: How To Start A Browse With The Last-Used QBE Query](#)

[Windows-Style List Box Sorting Revisited](#)

[Creating Filter Expressions The Easy Way](#)

[Checkboxes For Many-to-Many Relationships](#)

[Checkboxes For Many-to-Many Relationships: The Source Code](#)

[The Clarion Advisor: Sizing Windows](#)

[Write A Word Processor In Five Minutes](#)

[Running Totals: A Reader's Perspective](#)

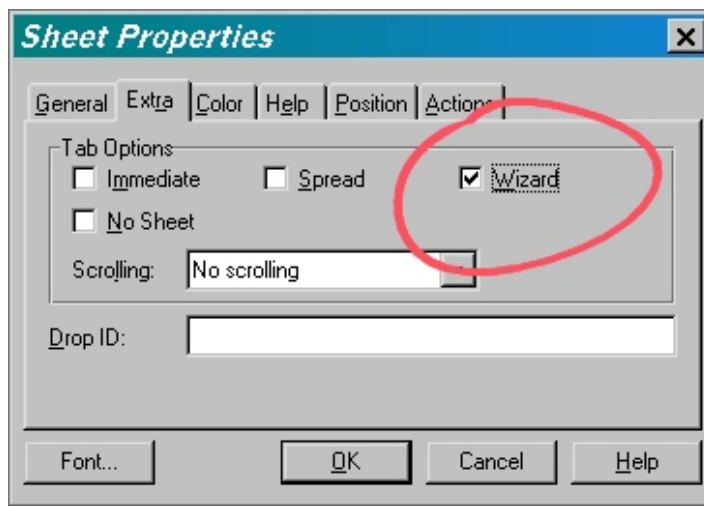
[Using Real Icons In The Listbox Header - Part 1](#)

[Using Real Icons In The Listbox Header - Part 2](#)

[Using Real Icons In The Listbox Header - Part 3](#)

[Advanced Reporting - Detail Filtering to Handle Multiple Relationships](#)

[Complex Filters - When parent records have no children](#)



**Figure 7. Sheet properties**

The advantage with this solution is that it is still possible to have a locator field which changes when a header selection has been made and a new sort key has been selected.

## The Browse Header Template

So much for coding the template; how do you use it?

For starters, the template consist of two extension templates:

- GlobalBrowseHeaderProperties
- BrowseHeader

You add the `GlobalBrowseHeaderProperties` template in the application's Global properties under the Extension and Control Templates menu. Here you can set the default values. For example it is not desirable that the header buttons ever receive focus, therefore the skip control check box is selected as a global condition; every time you use the `BrowseHeader` template, the global default condition is applied. This really cuts down the tedious work of setting up the default values for each button. Just be aware that this template *has* to be implemented first. If omitted you will soon find out when using the `BrowseHeader`. It will literally generate hundreds of errors to which you will have to answer OK.

In the `BrowseHeader` template you can set the more specific conditions for the selected column. For example, it is unlikely that you want the user to be able to select every column header button in the list box. To prevent this you can just select the disable property check box in the template for the particular column.

[Conditional Browse Formats](#)

[Advanced Reporting - Detail Filtering to Handle Multiple Relationships](#)

[Advanced Reporting 2 - Queues and Reports](#)

[Hand Coded View Templates in C4+ABC](#)

[Windows Styled Listbox Marking](#)

[Multiple Locators](#)

["Tab Popup" Template](#)

[Wrapping a Memo in a BrowseBox](#)

[Filtering Browsers on the Fly](#)

[Calling a Form - from a Java-Free Browse Box](#)

[Mike's Meanderings - Multi-Threaded Browsers and Forms](#)

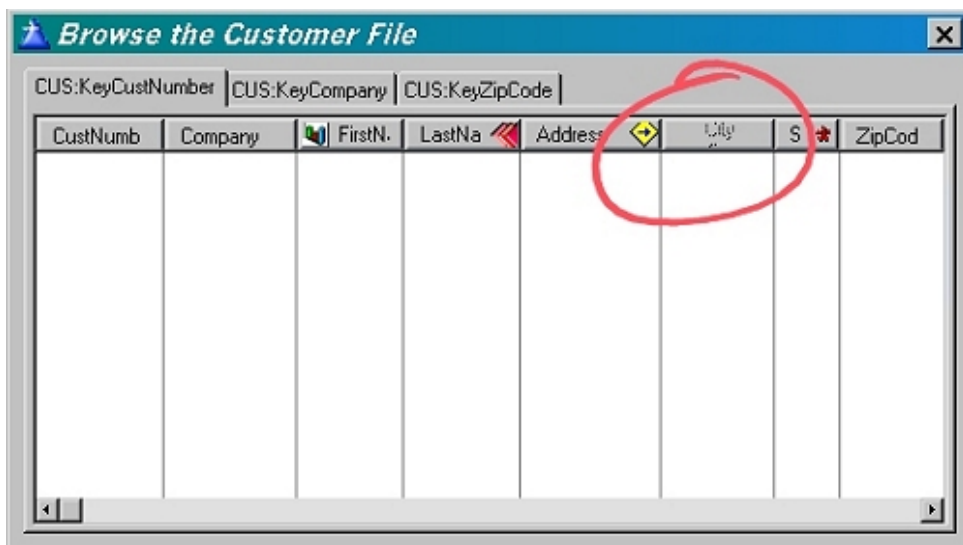
[Order! Order! Order in the Files!](#)

[Playing with the Browse Order](#)

## Known bugs

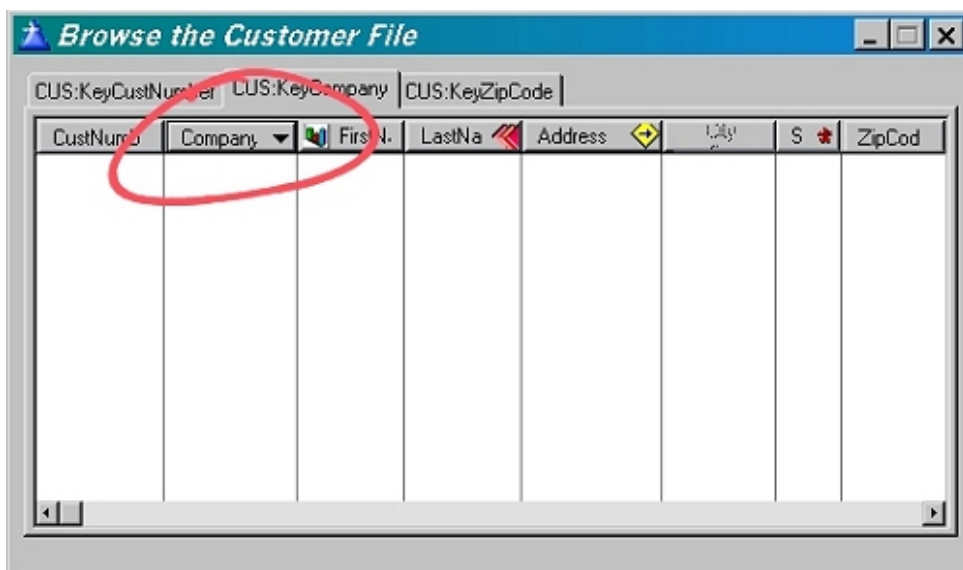
Yes, unfortunately there are still bugs, and I have probably only scratched the surface.

*Selecting the disable property:* This disables the header button as intended but it also ignores other settings for the particularly button as for example the left/right justification of the text.



**Figure 8. Disable property error**

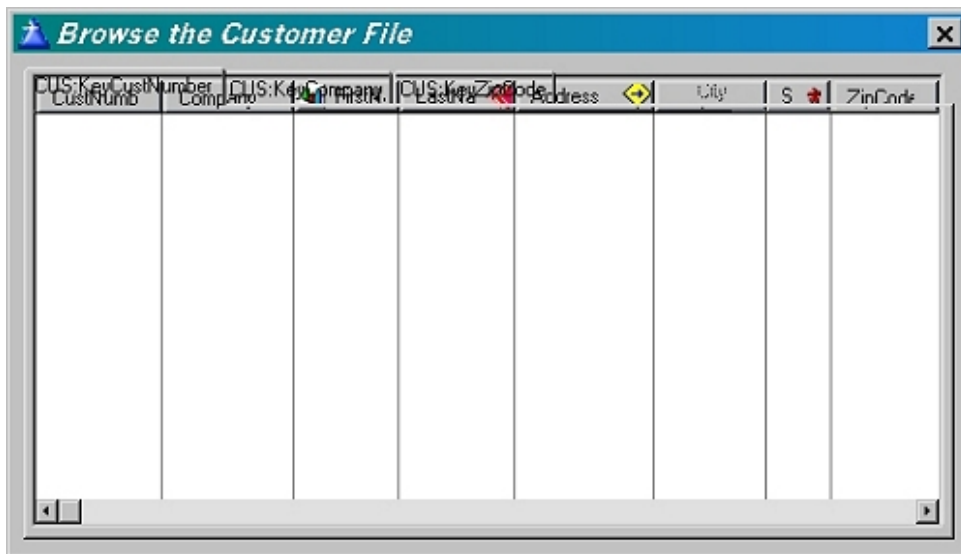
*Depressing a button:* When you select a button its size is reduced. It regains its original size when you select it again, or when you select another button.



**Figure 9. Depressed button error**

*Using the Clarion resize template:* When you use the Clarion resize template it seems as if the template assumes that there is a list box header; if there isn't, it just adds it to the list box.

So actually the list box becomes higher than intended.



**Figure 10. Clarion resize template error**

One workaround could be to make the list box height smaller at design time and resizing the window at start up.

*The button name:* Multiple words in the header must not be separated by spaces, instead you could write the name in one word e.g. `FirstName`. Although this is not a bug, but rather a button feature, it is not desirable in most cases.

### Where to go from here?

This template is far from finished and I have only tested it for those conditions which where applicable in my program. Some improvements might be:

- Fix the above bugs
- Add the possibility to select between normal button and radio buttons
- Make the template work when using edit-in-place
- Change width of columns on the border between the header buttons
- Include an embed button in the template
- Include a font button in the template

You can continue the list...

As you can see there is still room for a lot of improvement and, for that matter, testing, so let me know about any changes you make.

[Download the source](#)

[Steffen S. Rasmussen](#) has graduated in Computer Science from Copenhagen Business College. Since then he has worked as a programmer, system technician and network administrator, and is currently IT manager. Clarion is a quite a new language to Steffen since his only been working with it since January 2000. But what better way to learn it than by trying to teach others! Steffen has also set up a [web site](#) to collect as many examples of different user interfaces as possible to inspire Clarion developers.

## Reader Comments

[Add a comment](#)

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

[Home](#)   [COL Archives](#)
[Advanced Search](#)
[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)
[Home](#)
[Topics](#) > [News](#) > [ClarionMag 2001 News](#)
[Back Issues](#)
[COL Archives](#)

## Clarion News

[Advertise](#)

 Published 2001-11-21   [Printer-friendly version](#)
[Reader](#)
[Comments](#)

### [twTools 1.1](#)

twTools version 1.1 has been released. This new version includes a twRTFWriter, which you can use to generate RTF reports, view them automatically in a word processor, and print them with a call to your word processor. The RTF Writer does tables, with or without borders.

*Posted Tuesday, March 19, 2002*

[Log In](#)
[Membership/ Subscriptions](#)
[FAQ](#)

### [Gitano Build G Updates](#)

Clarion 5.507 (Build G) updates for various Gitano Software products are now available, including gCal, gCalc, gQ, gSec, gReg, and gNotes. This is a free download for registered users of these versions.

*Posted Tuesday, March 19, 2002*

[Privacy](#)
[Policy](#)
[Contact Us](#)
[PDFs](#)
[Freebies](#)
[Open Source](#)

### [CapFlash v3 Released](#)

CapFlash is a freeware extension template for processes; it will convert a file of all upper case to proper case (capitalize) with various options. Version 3 changes are courtesy of Robert A. Healy.

*Posted Tuesday, March 19, 2002*

### [INN Bio for 14-Mar-2002](#)

This week's INN Bio is the first featuring a British Clarionite, and another in Icetips' continuing series of presenters and others involved in the upcoming ETC conference. A magician, director, producer and writer, he has a rich and interesting resume in the show business world. In the Clarion world, we

### Related Categories

[ClarionMag 2001 News\(12\)](#)

### Related articles

[Clarion News - November 2001](#)
[Clarion News - January 2001](#)
[Clarion News - February 2001](#)
[Clarion News - March 2001](#)
[Clarion News - April 2001](#)
[Clarion News - May 2001](#)
[Clarion News - June 2001](#)
[Clarion News - July 2001](#)
[Clarion News - August 2001](#)
[Clarion News - September 2001](#)


know him as a Brit with a rather noticeable sense of humor, and he's also the official Clarion trainer for England. In his bio, he mentions being slapped in the face with a fish and OOP in the same sentence...

*Posted Friday, March 15, 2002*

### **[RPM C5.5.07 Update Available](#)**

RPM Version 5T is available for download. This is a free update for current licensed users.

*Posted Friday, March 15, 2002*

### **[EasyVersion 1.01 Released](#)**

New features in EasyVersion 1.01: Ability to use macros in a field "File Name" on the Tab "Version"; Correct formation of EXPORT (EXP) file name depending on macros used in the Target file name in the Project (standard templates do not make it). Fixes include changes to the name decrypting function and minor template changes.

*Posted Friday, March 15, 2002*

### **[EmailReport 2.1](#)**

Vivid Help has released EmailReport 2.1. You can now duplicate a report as a universal RTF attachment or send the report in this format. EmailReport 2.1 Legacy Template is released as well, and has RTF features similar to ABC version.

*Posted Friday, March 15, 2002*

### **[xAppWallpaper Manager v1.5](#)**

New in xAppWallpaper Manager 1.5: MultiInsert picture files in list now works normally; new version xPictureProwse; the input of registration information is simplified; fixed icon bug with xAppWall Manager compiled in LIB-mode. New demo and install available.

*Posted Friday, March 15, 2002*

### **[Buggy 3.1 Available](#)**

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

*Posted Friday, March 15, 2002*

### **[twTools For Clarion](#)**

twTools for Clarion has been released. This is a collection of objects to manipulate data. Most Clarion programmers are currently tied to the record, queue, and group structure,



which in many cases is very limiting due to its static nature. The objects in twTools are structureless for the most part. Imagine a "group" that has no fields, but fields can be added at run time. Now imagine an object that will store a collection of such dynamic groups. Consider the flexibility and power of being able to pass such objects around in your programs as one parameter, with no need to make new prototypes as your "groups" grow in structure. twTools is also the basis for the upcoming twMySQL product (currently in beta), which will allow you to access MySQL without file drivers or ODBC.

*Posted Monday, March 11, 2002*

### **[ConVic 2002 March 23-24](#)**

The ConVic 2002 Clarion get together will be held later this month, March 23/24 in Geelong, Victoria, Australia.

*Posted Monday, March 11, 2002*

### **[New Generation Of SetupBuilder Install Engines Test](#)**

The next SetupBuilder release comes with a new generation of install engines. SetupBuilder 4.02 (Beta) already uses the new engines, but the "runtime compression" feature is disabled. The current installer footprint is 141 KB. The new installer generation uses high-speed setup runtime compression to reduce the footprint to 97 KB (-31%)! The decompression library, installer library, and web-install technology is now packed in this small overhead of 97 KB. A test installer executable (105 KB) is now available. NOTE: The installer will not modify your system in any way! If you start the installation, it will copy a small dummy file into c:\Installer Test.

*Posted Monday, March 11, 2002*

### **[Clarion Skill Pool Web Site](#)**

The Clarion Skill Pool has now got its own web site, which will greatly increase its visibility on the search engines. To appear in this directory you don't need to be a Clarion Developer - the only requirement is that you can offer services which would be of interest to Clarion Developers. These could be web site design, marketing skills, subcontract programming etc.

*Posted Monday, March 11, 2002*

### **[Clarion 5.507 Released](#)**

SoftVelocity has posted the final release for Clarion 5.507.

There was one additional fix since the prior candidate release: there was a problem when deleting a control, extension or code template populated into window which has #LOCALDATA. If you deleted the template later, local data could become lost because the procedure could contain an incorrect pointer to the data.

*Posted Saturday, March 09, 2002*

### **[Free .NET Development Environment](#)**

Mark Sarson points out that a free .NET development environment, SharpDevelop, is available for those who would like to explore C# and VB. You will also need to download the .NET framework from Microsoft.

*Posted Thursday, March 07, 2002*

### **[Accounting ComCode Version 5.1 Released](#)**

A new Accounting release is now available for members.

*Posted Thursday, March 07, 2002*

### **[Taboga RTFWriter Library](#)**

Taboga Software has released version 1 of its RTFWriter library. This library allows you to construct RTF documents without knowing too much of the RTF specification. This is a freeware product. Features include: support for Microsoft Word 95, Microsoft Word 97, Microsoft Word 2000, Lotus WordPro, Word Perfect, TX TextControl and WordPad (Microsoft's Rich Text Control); Document and/or Section Format Control; full character formatting, specifically full font formatting, including size, style and color; paragraph formatting, including centering, justification, indentation, borders, color, tabs, etc.; FTF Tables support; headers and/or footers.

*Posted Thursday, March 07, 2002*

### **[Taboga Barcode Library 1.1](#)**

Taboga Software has released Barcode Library version 1.1. As previously indicated, the barcode library has become a commercial product with this release. The price is set at \$79.00. Symbologies include: Code39, Code39 Extended, Code 128, Code 128A, Code 128B, Code 128C, UCC128, UPCA, UPCE, EAN13, EAN8, Interleaved 2 of 5, and Code 2 of 5. You can use either the supplied template or handcode your projects. For those barcodes where the check digit is

optional, you may have the system determine the check digit and include it for you by setting up just one property. For those barcodes where the check digit is mandatory (UPC and EAN), you have the option of omitting the check digit, and have the system determine and include the check digit for you. For some symbologies you can change the ratio of the "wide" to "short" bars - i.e. in CODE 39 this is quite important to improve scanner efficiency. The library offers either automatic or custom control of how wide you want the barcode to be. You can even set up a barcode to print in an exact number of pixels per bar. Compatible with the Legacy Template chain (new in release 1.1)

*Posted Thursday, March 07, 2002*

### **[twTools for Clarion Pre-Release Demo](#)**

twTools for Clarion is nearing its first release. While some loose ends are tied together, you can download a demo version. The help file contains the documentation, and gives several complete programs as examples that you can compile and run.

*Posted Thursday, March 07, 2002*

### **[INN Bio for 6-Mar-2002](#)**

The Icetips News Network is pleased to feature another interesting Clarionite this week. An artist and a programmer, he has been using Clarion since Version 1.0. He mentions some of his experiences along the way, and offers an in-depth comparison of programming and art. He paid his board while living in Switzerland by working as a carpenter, and is now active in his town's government in Massachusetts. He's an obviously proud father, and enjoys beer, wine, and carrot juice. And for photos, think "H"s... Horses and Hawaii.

*Posted Thursday, March 07, 2002*

### **[PDF-XChange & Tools Last Chance Pricing](#)**

Later this week/early next Tracker Software will be releasing Version 2 Of the PDF-XChange API Library and Royalty Free Printer drivers. There are many, many improvements, including over 60 new library functions and massively improved file optimization/compression. Threading has also been improved to allow synchronous and asynchronous printing without file corruption. You can append to file, convert images to PDF (BMP, GIF, JPEG, PNG, TIFF and WMF) and extract PDF images to all the aforementioned formats

with the exception of WMF. This is a free upgrade for existing users - but please note the Version 2 API/drivers template and classes for Clarion are a little further behind - but not too far. In addition Tracker Software will also be releasing the extended template/classes for PDF-Tools with PDF File merge, page extraction, image to and from PDF etc. this weekend. Once released, Tracker will be altering pricing as follows - so please take advantage of the current pricing and free upgrade to version 2 before the increase if you are close to buying: PDF-XChange - Full suite - Full API, Royalty Free Drivers (all Windows versions) and PDF-Tools, Template/Classes, currently \$699.00 increasing to \$899.00; PDF-XChange SDK (all Windows versions) Royalty Free Printer Drivers and Template/Classes, currently not sold separately, new price \$699.00; PDF-Tools API/Template/Classes, currently \$449.00, increasing to \$499.00. Next on the development menu and to be released in the coming weeks: Twain Scan to PDF; Create PDF and simple FTP/HTTP/MAPI/SMTP functionality; PDF to ASCII - retrieve all the text from a PDF and save or search for required values or indexing purposes etc.

*Posted Thursday, March 07, 2002*

### **BigTamer(tm) Templates And Utilities**

Earl Coker and Bo Schmitz have teamed up to produce the BigTamer(tm) Templates and Utilities. There are quite a few templates and example apps. Features include: filtering; exporting; embed management; image handling; template creation; color coding. Possible applications include: managing the conversion of a Legacy app to an ABC app with respect to embed point differences; learning the Clarion template language; documenting existing embeds and improving reuse.

*Posted Thursday, March 07, 2002*

### **GUI.IT Template Beta Demo**

Langaard Software has released a demo application for the GUI.IT templates. Features include: dockable toolbars; assign work areas for your windows; individual resizing of all windows in your application (as OE Express does); automatic opening of procedure in work area on startup; window enhancement on getting focus; support for all types of controls in docked toolbars. During the beta get the

templates for \$38; after release the price will be \$99.

*Posted Thursday, March 07, 2002*

### **[SysCalPack Bundle Available](#)**

A new SysCalPack bundle is now available from solid.software. This bundle includes both calendar-related products: SysDTP, a wrapper for the date and time picker control, and SysMonthCal, a wrapper for the month calendar control. The price of the new bundle is US\$59, for a savings of \$19. users of SysDTP OR SysMonthCal can update their licenses for \$20. Existing users of both SysDTP AND SysMonthCal get a \$20 discount when buying any other solid.software products. SysCalPack is also available at ClarionShop.

*Posted Thursday, March 07, 2002*

### **[New Bundles From Gitano Software](#)**

Effective immediately every Gitano Software bundle purchased will include a free copy of PDF Manager to manage all your PDF Files (such as PDFs from Clarion Magazine) as well as a copy of G-BUDDY, a Clarion Developers Swiss Army Knife with a little tool for every developer. Gitano's new upgrade policy allows you to upgrade from any Gitano Software product to a bundle that includes that product, or from one bundle to another one, for the retail price less the amount you paid for the original purchase. If you own any product in the bundle Gitano will discount the price you paid for the product from the cost of the bundle.

*Posted Thursday, March 07, 2002*

### **[Legacy Apps And MySQL](#)**

Kristian Hyllestad of SPINE AS reports his company has posted several pages of information on using MySQL with Clarion legacy templates. Included are instructions on making necessary changes to two template files. Also available are PHP (for speed comparison) and Clarion demo applications. Both may be tested online.

*Posted Thursday, March 07, 2002*

### **[Linder SetupBuilder 4.01 Receives Softmag's Editor's Choice](#)**

Linder Software's SetupBuilder 4.01a Standard Edition was recently reviewed by Thomas Thorp, editor of Softmag, Norway's leading online computer publication, and given the

Editor's Choice Award with the highest rating, saying it is "...one of the best programs to make software installations".

*Posted Tuesday, March 05, 2002*

### **CPCS Releases Excel/ASCII Output Addon**

CPCS has announced the release of the CPCS Excel/Ascii Output Addon. This new product allows you to output precisely formatted, columnized, CSV and ASCII files from your CPCS reports, for direct import into Microsoft Excel® and other applications that import CSV and TXT files. Optionally, you can include column headers in your output files. Unlike the "TextOut" capability in the main CPCS product (which outputs all info on every page of a report), this addon allows you to precisely define the data to be output based on variables, constants, and/or report controls. This product requires Clarion 5b or 5.5 Gold or later.

*Posted Tuesday, March 05, 2002*

### **Linder SetupBuilder 4.02 Web Edition Beta 1**

The Beta-1 release of SetupBuilder 4.02 Web Edition is now available. SetupBuilder's Web-Install technology allows you to very easily deploy and update your applications over the Internet. The installation runs as a stand-alone application and uses the WINSOCK API to download the necessary files required for the installation. SetupBuilder does not rely on browser plug-ins or ActiveX controls. The purpose of Beta 1 is to test the Web-Install communication module. Beta 2 will provide the Web Update feature and multi-patch capability (both features are already code complete, but require a rock-solid communication module). All registered beta testers will receive the password to install Web Edition within the next 12 hours. Registered Web Install customers will receive the registration code within the next 12 hours.

*Posted Tuesday, March 05, 2002*

### **DOS Printer Version 6 Released**

David Beggs has released DOS Printer version 6, a C5b ABC program designed to prolong the life CPD2.1 programs. It sits in the system tray and monitors for the presence of a specified text file. When it finds it, it converts the ANSI line type characters to their nearest courier equivalent and prints it to a pre-set windows printer. If you have DOS clients who have trouble printing to USB printers, or non-dos compatible

printers, or who would like to print to fax or PDF type drivers then check it out. New in version 6: You can now select any file and print it (without deleting it afterwards); DOS Printer now uses the paper size of the selected printer (it used to just print on a4). Source and distribution licenses available.

*Posted Tuesday, March 05, 2002*

### **[Gitano Requires Paid Email Accounts For Online Purchase](#)**

Gitano Software is implementing a policy change banning purchases via free email addresses such as hotmail.com. This change is required because of the incidence of fraud related to free email accounts. If you have only a free email account and still want to purchase, contact Gitano Software for further information.

*Posted Friday, March 01, 2002*

### **[EasyDocker 1.0](#)**

Ingasoftplus has released EasyDocker 1.0, a class and template which provides you with the ability to include floating toolbars into your Clarion application. You can start procedures, mimic global menu items and implement standard windows behavior in the floating toolbars. Also you can transfer your standard "static" global toolbar functionality into the floating toolbar with one click. Clarion 5.0b or 5.5, ABC or Legacy, 32-bits only.

*Posted Friday, March 01, 2002*

### **[gCal 3.0 Update Now Available](#)**

An update for gCal 3.0 is now available to all registered users. This update fixes the compile errors with the new calendar control and contains the icons omitted in the previous installation.

*Posted Friday, March 01, 2002*

### **[xNotes v1.3 Released](#)**

New in SealSoft's xNotes release 1.3: SetCheckEnabled method to enable/disable check reminders during a long process; minor bugfix regarding use of xTipOfDay together with xDataBackupManager, xNotes, xAppWalpaper and xAppWalpaper Manager. New demo and install available.

*Posted Friday, March 01, 2002*

### **[xTipOfDay v1.4 Released](#)**

SealSoft's xTipOfDay is now in release 1.4. New features include: starting xTipOfDay after splash window; minor bugfix regarding use of xTipOfDay together with xDataBackupManager, xNotes, xAppWalpaper and xAppWalpaper Manager. New demo and install available.

*Posted Friday, March 01, 2002*

## **Reader Comments**

[Add a comment](#)

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



[Home](#) [COL Archives](#)[Advanced Search](#)[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)[Home](#)[Back Issues](#)[COL Archives](#)[Advertise](#)[Reader](#)[Comments](#)[Call for](#)[Articles](#)[Log In](#)[Membership/](#)[Subscriptions](#)[FAQ](#)[Privacy](#)[Policy](#)[Contact Us](#)[PDFs](#)[Freebies](#)[Open Source](#)[Topics](#) > [Design & Development](#) > [Conversions](#)

## Migrating The Inventory Application To SQL Server (Part 4)

**by Ayo Ogundahunsi**Published 2002-03-07 [Printer-friendly version](#)

In the [most recent article](#) in this series I focused on Data Migration using Microsoft SQL Server Data Transformation Services (DTS). In all the examples I presented, I didn't alter any of the data. However, when migrating from a flat-file system to an SQL environment, you can never have an efficient SQL application if you are only comfortable with a straight conversion, where data access and updates still follow the logic of a flat-file system. On the other hand, if you intend to make the application more SQL-like, you might have to do some data manipulation, column filling and table splitting to guarantee a more efficient system.

In this article, I will show you a different method of moving data into SQL Server. I will also demonstrate how the data can be manipulated and converted to a more efficient relational database model.. To get the most out of this information you should also read my previous articles in this [series](#).

### A new database schema

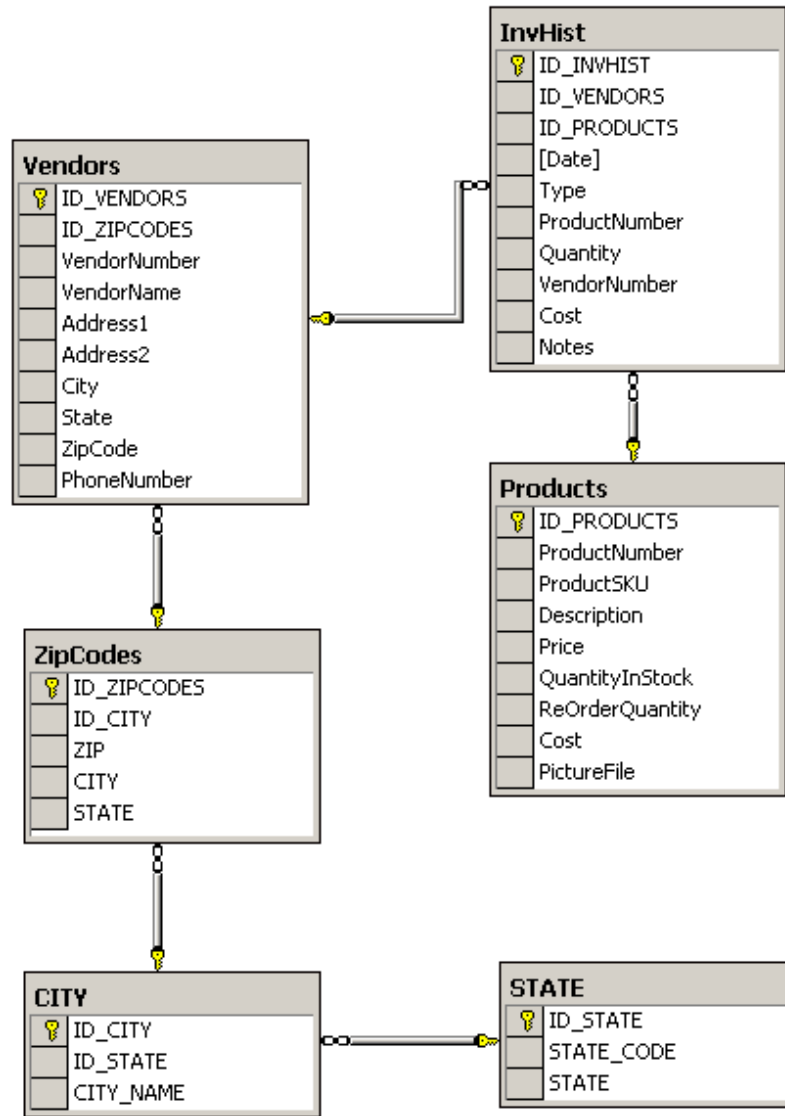
In my [last article](#) I suggested splitting part of the database in order to have two new tables: `STATE` and `CITY` (this is one example of normalization). Figure 1 shows how these changes will be reflected in the database structure.

### Related Categories

[SQL\(34\)](#)[Conversions\(17\)](#)[Database Design\(16\)](#)[Database Design\(16\)](#)

### Related articles

[The SQL Answer Cowboy](#)[The SQL Answer Cowboy Answers!](#)[How To Convert Your Database To SQL](#)[The Novice's Corner - Designing Databases](#)[The SQL Answer Cowboy](#)[The SQL Answer Cowboy](#)[Stephen Mull's Guide To Converting To MS-SQL](#)[Presenting Many-To-Many Relationships](#)[Presenting Many-To-Many](#)



**Figure 1. New Schema for Normalization (1<sup>st</sup> Stage)**

Notice `dbo.CITY` and `dbo.STATES` have been added, but for now the columns `dbo.ZIPCODES.CITY` and `dbo.ZIPCODES.STATE` still exist in `dbo.ZIPCODES`. The `dbo` prefix specifies the database owner. I will be qualifying tables in SQL Server occasionally this way in order to differentiate them from Topspeed files.

When converting a flat file system to SQL, most developers start the design of the database from the Clarion Dictionary and then export to SQL. For database maintenance, most developers will make changes in Clarion and then try to use the synchronizer to push the updates to SQL Server. I prefer to make the changes in SQL Server and then use the synchronizer to push the updates to Clarion. (Actually, I make the changes in the Clarion Dictionary manually most of the time.)

You can get the script `State_City.sql` that creates the `STATE` and

[Relationships: Part 2](#)

[The SQL Answer Cowboy](#)

[The SQL Answer Cowboy](#)

[Using Clarion With MySQL - Part 1](#)

[Using Clarion With MySQL Part 2](#)

[Multi-Platform Development: Clarion, MySQL, and Linux Part 3](#)

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

[MySQL/MyODBC Update](#)

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

[Legacy to ABC: There is Another Way!](#)

[Legacy to ABC: There is Another Way! Part 2](#)

[Legacy to ABC: There is Another Way! Part 3](#)

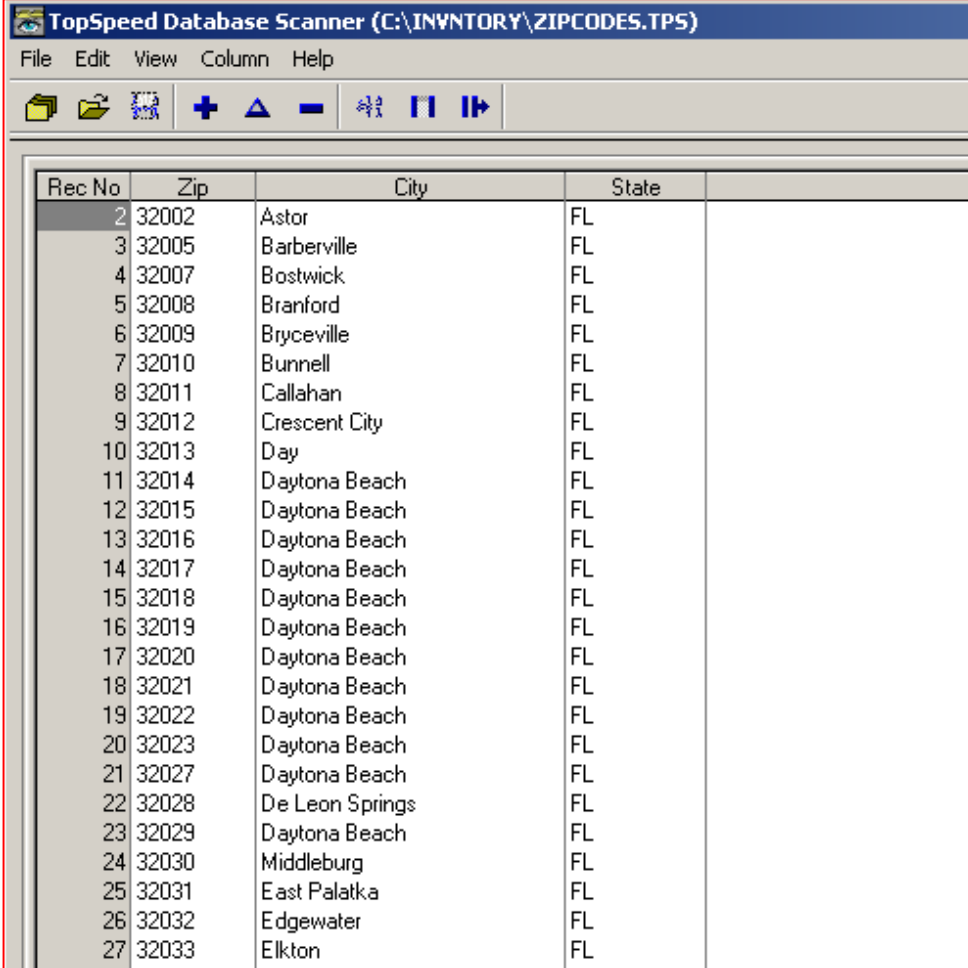
[Five Rules for Managing Complexity: Part 1](#)

[Five Rules for Managing Complexity:](#)

CITY tables as part of the downloadable source at the end of this article. It is important you run this script before you start any conversion.

## Zipcodes

If you take a look at the TopSpeed ZIPCODES file, C:\INVNTORY\ZIPCODES.TPS, you will notice that the State field data is abbreviated, for example 'FL' for Florida, as shown in Figure 2. However, I need to be able to get the full name of a state.



Rec No	Zip	City	State
2	32002	Astor	FL
3	32005	Barberville	FL
4	32007	Bostwick	FL
5	32008	Branford	FL
6	32009	Bryceville	FL
7	32010	Bunnell	FL
8	32011	Callahan	FL
9	32012	Crescent City	FL
10	32013	Day	FL
11	32014	Daytona Beach	FL
12	32015	Daytona Beach	FL
13	32016	Daytona Beach	FL
14	32017	Daytona Beach	FL
15	32018	Daytona Beach	FL
16	32019	Daytona Beach	FL
17	32020	Daytona Beach	FL
18	32021	Daytona Beach	FL
19	32022	Daytona Beach	FL
20	32023	Daytona Beach	FL
21	32027	Daytona Beach	FL
22	32028	De Leon Springs	FL
23	32029	Daytona Beach	FL
24	32030	Middleburg	FL
25	32031	East Palatka	FL
26	32032	Edgewater	FL
27	32033	Elkton	FL

**Figure 2. Original Zip Code file**

Also, this zipcodes file contains about 3,924 records, whereas you can get a file containing about 48,254 records from [Steve Parker's site](#). In order to avoid confusion with the original zip code file shipped with Clarion, I will rename the downloaded file to ZIPS.TPS.

A close look at the contents of this downloaded file shows that it also contains only the abbreviation of states. While this is okay for something like a mailing label e.g. MA 01001, it is inadequate for

[Part 2](#)

[Five Rules for Managing Complexity: Part 3](#)

[Five Rules for Managing Complexity: Part 4](#)

[Five Rules for Managing Complexity: Part 5](#)

[Converting PowerBrowse from Legacy to ABC](#)

[Implementing SELECT DISTINCT in a TPS Database](#)

[Data Is Executed Policy - A Case Study](#)

[Information Systems Planning](#)

[The Clarion Advisor: Locating Records With PROP:SQLFilter](#)

[Introduction To SQL: Part 1](#)

[Introduction To SQL: Part 2](#)

[Introduction To SQL: Part 3](#)

[Is SQL A Real Standard?](#)

[Introduction to SQL - Part 4](#)

[A Column by](#)

proper report generation.

Rec No	City	State	Zipcode
2	Agawam	MA	01001
3	Amherst	MA	01002
4	Amherst	MA	01003
5	Amherst	MA	01004
6	Barre	MA	01005
7	Belchertown	MA	01007
8	Blandford	MA	01008
9	Bondsville	MA	01009
10	Brimfield	MA	01010
11	Chester	MA	01011
12	Chesterfield	MA	01012
13	Chicopee	MA	01013
14	Chicopee	MA	01014
15	Chicopee	MA	01015
16	Chicopee	MA	01016
17	Chicopee	MA	01017
18	Chicopee	MA	01018
19	Chicopee	MA	01019
20	Chicopee	MA	01020
21	Chicopee	MA	01021
22	Chicopee	MA	01022
23	Cumington	MA	01026
24	Easthampton	MA	01027
25	East Longmeadow	MA	01028

**Figure 3. Downloaded Zip Code file**

Note that ZIPCODES.TPS also contains some non-US Postal/Zip codes, and I have to retain these when I attempt to merge ZIPCODES.TPS and ZIPS,TPS.

I have itemized a list of operations needed to reconstruct this schema:

- Merge ZIPS.TPS (48,254 records) into ZIPCODES.TPS (3,924 records) while retaining non-US zip codes.
- Where zip codes match, use City field from ZIPS.TPS.
- Auto-increment ID\_ZIPCODES in dbo.ZIPCODES. This will now be the primary key of this table and SQL Server will set the value automatically on inserts, since this column was declared with an IDENTITY property. Whatever value is updated will have a relationship with the VENDOR table as explained in the next point.
- The column ID\_ZIPCODES in dbo.VENDORS will be updated with the corresponding ID\_ZIPCODES from dbo.ZIPCODES.

The last item does not have to be done at this part of the conversion, and will not be addressed in this article.

### Linked servers

[Any Other Name Is Not A Data Element](#)

[Avoid My SQL Mistakes!](#)

[Migrating The Inventory Application To SQL Server \(Part 1\)](#)

[Migrating The Inventory Application To SQL Server \(Part 2\)](#)

[Migrating The Inventory Application To SQL Server \(Part 3\)](#)

[Normal Files, Normal Display](#)

[MySQL: InnoDB Tables And Transactions \(Part 1\)](#)

[MySQL: InnoDB Tables And Transactions \(Part 2\)](#)

[Using SQL Server's Data Transformation Services \(DTS\)](#)

[A Template For Copying Fields Between Files](#)

[Migrating The Inventory Application To SQL Server \(Part 4\)](#)

[Simply Clarion - A Simple Deselect Control Template](#)

In my [previous article](#), I briefly mentioned *linked servers*. A linked server is virtual server created by a link from an SQL Server environment to an OLE DB data source. This link can be a relational database or columnar data in a flat file. It can even be an Excel spread sheet as you will soon see. See [MSDN](#) for more details on linked servers.

Included in the source download at the end of this article is the Excel spreadsheet State.xls containing all the states in the US. Note that there are other ways of importing the data contained in Excel file into SQL Server. You can export the spreadsheet to a text file and subsequently import it to SQL Server by using the `BULK INSERT` command in Transact SQL (Transact SQL is the dialect of SQL specific to SQL Server), or, you can use the `BCP` command. `BCP` is a command line utility for importing data into a SQL Server Table, or exporting the rows from a SQL Server table to a text file. You can also use Data Transformation Services (DTS) to import the Excel file as I demonstrated in my previous article.

But the most exciting way is to make the Excel file a linked server. This allows you to manipulate the Excel file like relational data, and even create a join (a `JOIN` helps you to return a result set whose columns can be a combination of two or more tables) or a View that includes the Excel data. After downloading STATE.XLS, copy it to C:\INVNTORY, or whichever directory you have been using for data conversion.

## The Excel spreadsheet

STATE.XLS contains the names of the 50 United States together with their two-character codes. In order for me to be able to access the cells in the Excel spread sheet, it is mandatory that I define a range of the data to be used in whatever query is required. This I have labeled `US_STATES`, and is from A1:B52. The first row contains the field names, since any query sent from SQL Server needs to understand what column is to be returned. Note that the first row is *always* returned as the column name.

[Understanding Btrieve and Scalable SQL4](#)

[Simply Clarion - Converting Simple Templates to ABC: Part 1](#)

[Adding Rules to the C4 Conversion Wizard](#)

[Further Understanding Btrieve - and Scalable SQL4](#)

[Simply Clarion - A Simple Template Conversion to ABC](#)

[Upgrading to Client/Server - using Oracle Accelerator](#)

[Simply Clarion - Converting 2.003 to C5 ABC - Part 1](#)

[Simply Clarion - Lookups, Aliases, and RI - Part 1](#)

[Simply Clarion - Lookups, Aliases and RI - Part 2](#)

[Simply Clarion - Lookups, Aliases and RI - Part 3](#)

[Simply Clarion - Lookups, Aliases and RI - Part 4](#)

The screenshot shows a Microsoft Excel spreadsheet titled 'state.xls'. The spreadsheet has two columns: 'STATE' and 'CODE'. The rows are numbered 1 through 20, corresponding to the US states and District of Columbia. The data is as follows:

	A	B	C	D	E	F
1	STATE	CODE				
2	Alabama	AL				
3	Alaska	AK				
4	Arizona	AZ				
5	Arkansas	AR				
6	California	CA				
7	Colorado	CO				
8	Connecticut	CT				
9	Delaware	DE				
10	District of Columbia	DC				
11	Florida	FL				
12	Georgia	GA				
13	Hawaii	HI				
14	Idaho	ID				
15	Illinois	IL				
16	Indiana	IN				
17	Iowa	IA				
18	Kansas	KS				
19	Kentucky	KY				
20	Louisiana	LA				

**Figure 4. Excel Spreadsheet.**

## Creating the Linked Server

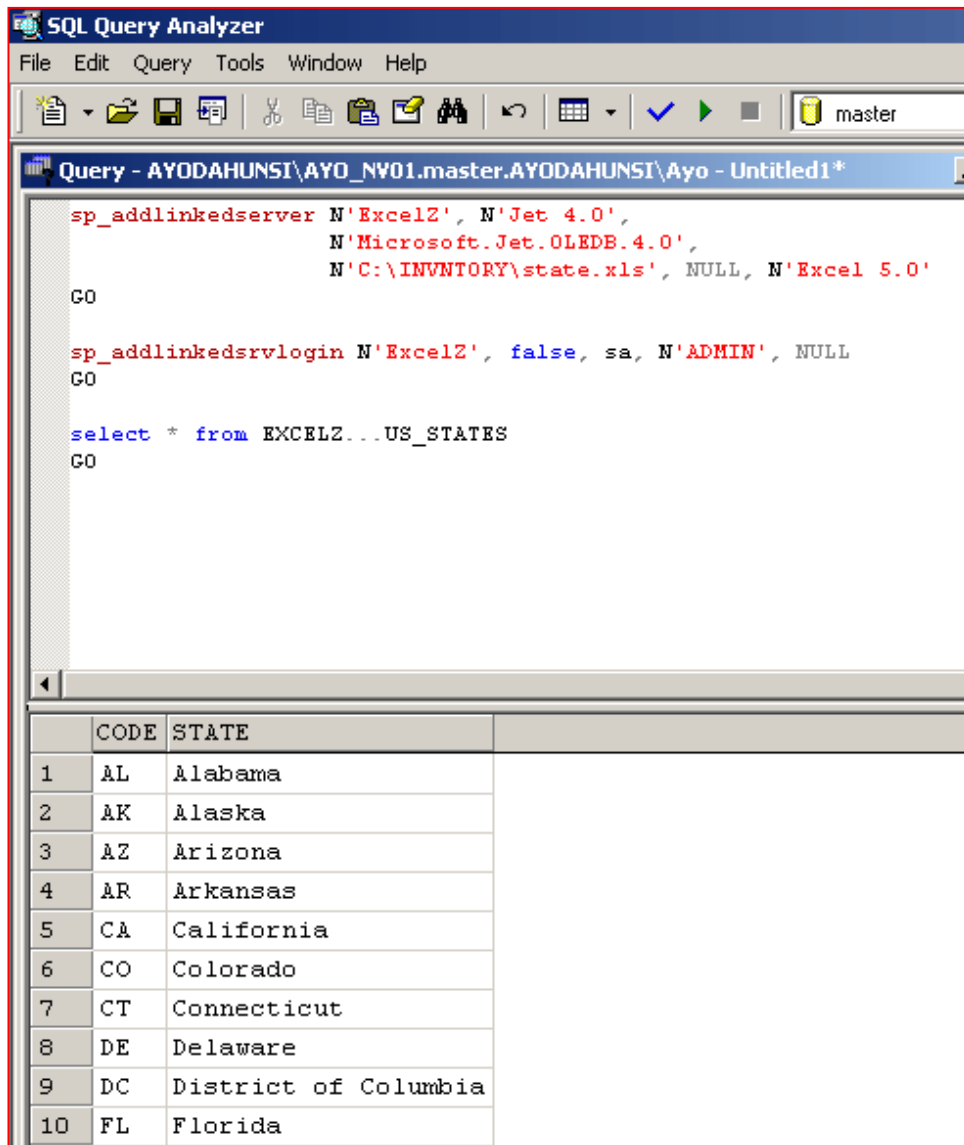
Creating a linked server is quite simple. You can do this from SQL Server's Enterprise Manager, or you can execute SQL to create it. I will use SQL rather than the easy way of using the Enterprise Manager. Below is the SQL code to create a linked server, add a login to the linked server, and then use `SELECT` to list the rows in the STATE.XLS. Copy this code to the Query Analyzer, and execute it.

```
sp_addlinkedserver N'ExcelZ', N'Jet 4.0',
    N'Microsoft.Jet.OLEDB.4.0',
    N'C:\INVENTORY\state.xls', NULL, N'Excel 5.0'
GO
sp_addlinkedsrvlogin N'ExcelZ', false, sa, N'ADMIN', NULL
GO
select * from EXCELZ...US_STATES
GO
```

In this SQL code, you will notice I preceded the string parameters with the `N` character. This means I am passing Unicode constants, and the parameters are interpreted within the SQL engine as Unicode data. On the other hand, I can remove the `N`s and the code will still work fine, but I will be passing string constants which are going to be evaluated with the engine using a code page.

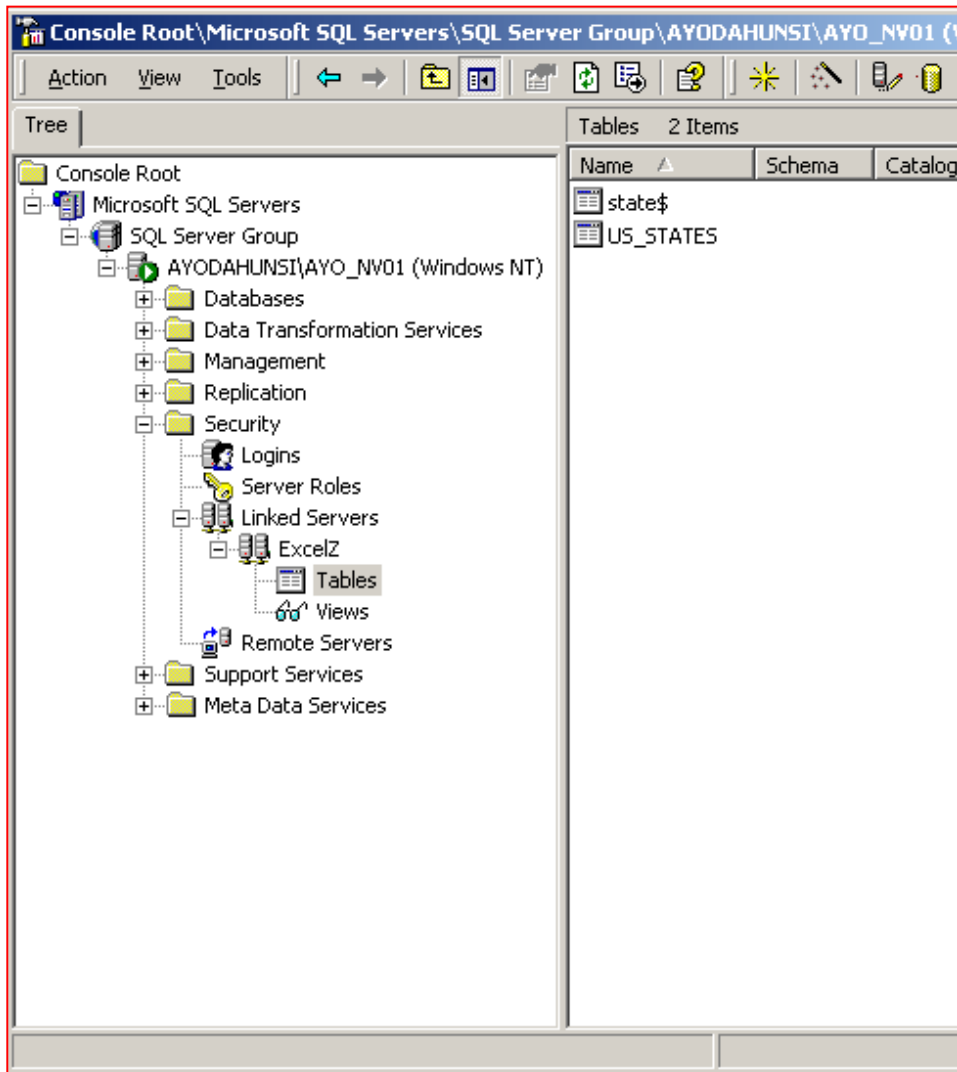
Unicode is necessary for internalization of applications

Figure 5 shows what the output will look like. Because there are state codes, you can create a JOIN/VIEW on the codes to ZIPCODE.TPS.



**Figure 5. Creating a Linked Server**

After you have created the Excel Linked server, you will see it appear under the Security section (See figure 6) of the SQL Server Group.



**Figure 6. Created Linked Server in Enterprise Manager**

I have used two stored procedures, `sp_addlinkedserver`, and `sp_addlinkedserverlogin`. The first one creates the linked server, and the second one helps to ensure that your local instance of SQL Server can have access to the linked server.

### Updating the STATE table

The first step in table updates for database restructuring has to do with `dbo.STATES`. Since I already have direct access to the Excel file, I can use an `INSERT` command to fill this table. What I want to do is to retrieve data from the Excel file and insert into `dbo.STATES` all in one go; this is what SQL is good at, working in a set-oriented way. The conventional (flat file) method is to loop through the data one row at a time. This supports my earlier comment about the need to restructure updates in a legacy system if you want to reap the benefits of migrating to SQL.

The script below will fill `dbo.STATES` with data from `STATE.XLS`. Remember to run the script `state_city.sql` first so that



`dbo.STATE` exists.

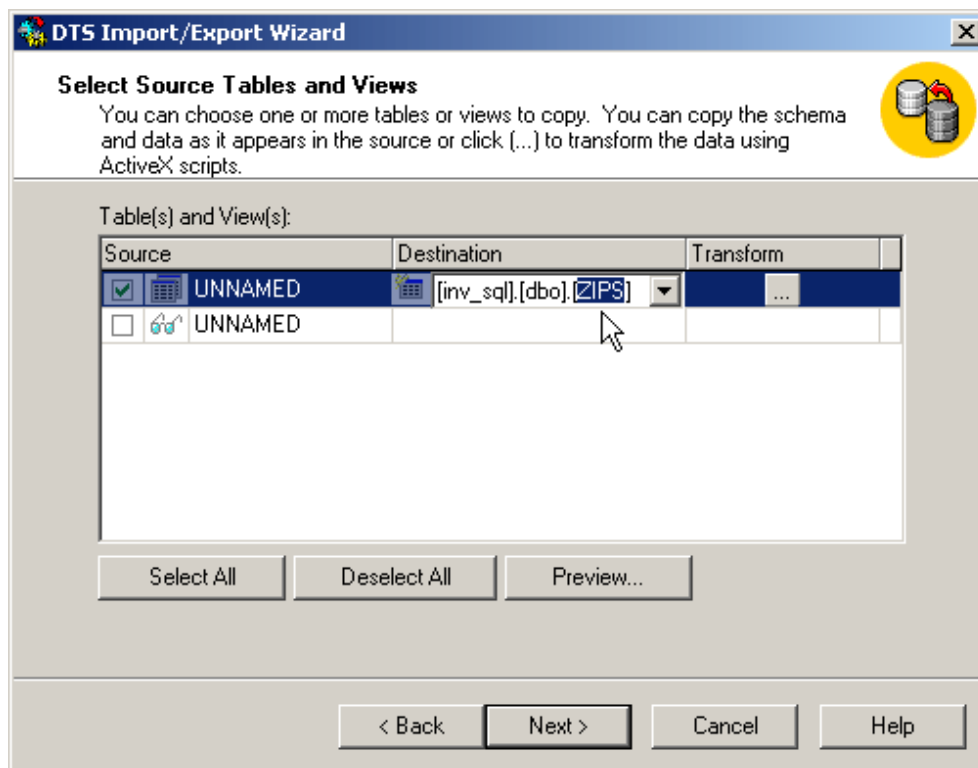
```
INSERT dbo.STATE (STATE,STATE_CODE)
SELECT STATE, CODE FROM EXCELZ...US_STATES
```

Note that the second line returns a result set in the order of the columns being inserted into the first line.

## Moving zipcode data to SQL Server

The list of cities and the zip codes is in ZIPCODES.TPS and ZIPS.TPS (this is the downloaded file). It might have been better to create another linked server to ZIPS.TPS, but as I mentioned in my last article, the Topspeed ODBC driver always hung my system whenever I tried doing anything complicated. For all I know, this might be a restriction on the read-only version I am using. So, I will use DTS and move the data in ZIPS.TPS into a staging area where I can then manipulate it.

Following the steps outlined in my article on using [DTS](#), I will move the contents of ZIPS.TPS into `dbo.ZIPS`. As `dbo.ZIPS` is non-existent in the database, I have to ensure that DTS is able to create it prior to data import.



**Figure 7. Naming destination table**

As shown in Figure 7, I named the destination table `ZIPS`. After I execute the DTS package, `ZIPS` contains a total of 48,253 records.

## Updating the CITY table

I need to perform a series of SQL tasks before I can populate `dbo.CITY`. The first thing I need to do is to logically combine the two zipcode tables, `dbo.ZIPCODES` and `dbo.ZIPS`, in a VIEW and then extract city information from here. At this stage, I need to create a logical, rather than a physical combination due to the fact that I have to populate the foreign key `dbo.ZIPCODES.ID_CITY` with the matching IDs expected to be in `dbo.CITY`. Unfortunately, I cannot do this now because I intend to populate the `CITY` table from the zipcodes tables. My planned stages of conversion are:

1. Create a VIEW that combines `dbo.ZIPCODES` and `dbo.ZIPS` into one result set. This View will contain the same rows expected to be in the final version of `dbo.ZIPCODES`. Below is the script that creates the view. The `UNION` keyword combines the output of two result sets into one single result set.

```
CREATE VIEW ALLZIP AS
/*
    Rows in dbo.ZIPCODES that do not have duplicates in
    dbo.ZIPS retrieved
*/
SELECT ZIP, CITY, STATE
FROM ZIPCODES
WHERE ZIP NOT IN
    (SELECT ZIPCODE FROM ZIPS)
UNION
/*
    Rows in dbo.ZIPS that do not have duplicates in
    dbo.ZIPCODES retrieved
*/
SELECT ZIPCODE, CITY, STATE
FROM ZIPS
WHERE ZIPCODE NOT IN
    (
-- Query to return all rows dbo.ZIPCODES.ZIP
    SELECT ZIP FROM ZIPCODES
    )
)
```

2. Create a JOIN combining the `ALLZIP` view with `dbo.STATE`. This will be used for populating `dbo.CITY`. Cut and paste the code below into Query Analyzer, and run to update the `CITY` table.

```
INSERT dbo.CITY (ID_STATE,CITY_NAME)
SELECT dbo.STATE.ID_STATE, dbo.ALLZIP.CITY
FROM dbo.ALLZIP LEFT OUTER JOIN
    dbo.STATE ON dbo.ALLZIP.STATE = dbo.STATE.STATE_CODE
```

3. Export the contents of `ALLZIP` to a temporary file. This is necessary because the `ALLZIP` view is built by combining two tables in a non-efficient way, and moreover, the `ZIPS` table does not have a primary key or any indexes, so performance is poor. At the same time, the data returned by this view is not static and is dependent on what is in `dbo.ZIPCODES` and `dbo.ZIPS`, so any updates to `dbo.ZIPCODES` will affect what is in the view. Below is the code to

## copy the contents of the view to a temporary file

```
SELECT * INTO #temp_zip FROM allzip
```

The `SELECT INTO` command will create a new table having the same structure as the `FROM` table, and fill it up with row of the `FROM` table. Starting a table name with a single hash "#" indicates the table is created as a local temporary table. This table exists during the life of a connection. Once a connection is terminated, SQL Server automatically deletes the table. Also the table cannot be seen from another connection. This is different from a global temporary table, which starts with two hash signs (e.g. ##temp\_zip) and is visible to other connections, but is deleted once all connections referencing the table have been disconnected.

4. The next thing to do is to delete all records from `dbo.ZIPCODES`, and fill it with rows from `#temp_zip`. I also need to correct the length of one of the columns in `dbo.ZIPCODES`.

The column `dbo.ZIPCODES.CITY` is 25 characters in length whereas, the corresponding field `dbo.ZIPS.CITY` is 28 characters. This I need to take care of before I start transferring any from `#temp_zip` since this temporary file is created from the `ALLZIP` view, which is created from a combination of `dbo.ZIPS` and `dbo.ZIPCODES`.

The script below is in four parts (separated by the `GO` keyword). The first part checks for and removes the Foreign key constraint `dbo.ZIPCODES` has with `dbo.VENDORS` (this is added back in the last part of the script). The second section `DROPS` the table; this effectively removes all the rows, as intended. The third section recreates the table only, but this time the size of the `STATE` column is increased to 30.

```
if exists
(
select * from dbo.sysobjects
  where id = object_id(N'[dbo].[FK_ZIPCODES_VENDORS]')
    and OBJECTPROPERTY(id, N'IsForeignKey') = 1
)
ALTER TABLE [dbo].[Vendors] DROP CONSTRAINT
  FK_ZIPCODES_VENDORS
GO
if exists
(select * from dbo.sysobjects
  where id = object_id(N'[dbo].[ZipCodes]') and
    OBJECTPROPERTY(id, N'IsUserTable') = 1
)
DROP TABLE [dbo].[ZipCodes]
GO
CREATE TABLE [dbo].[ZipCodes]
(
```

```

        [ID_ZIPCODES] [int] IDENTITY (1, 1) NOT NULL,
        [ID_CITY] [int] NULL,
        [ZIP] [varchar] (10) NULL,
        [CITY] [varchar] (30) NULL,
        [STATE] [varchar] (2) NULL
    )
ON [PRIMARY]
GO
ALTER TABLE [dbo].[ZipCodes] WITH NOCHECK ADD
    CONSTRAINT [PK_ZIPCODES] PRIMARY KEY CLUSTERED
    (
        [ID_ZIPCODES]
    ) ON [PRIMARY]
GO

```

5. After `dbo.ZIPCODES` has been modified, the next thing to do is import rows from `#temp_zip`.

```

INSERT dbo.ZIPCODES (ZIP,CITY,STATE) -- Fill from #temp_zip
SELECT ZIP, CITY, STATE
FROM    dbo.#temp_zip

```

6. After this I need to map the IDs in `dbo.CITY` to `dbo.ZIPCODES` and update `dbo.ZIPCODES` accordingly. If you are coming from a procedural programming background, what readily comes to mind as a solution is `LOOPING` through all the rows (records), of which there are about 44,000 in the table. Processing many records this way is rather inefficient, which is why the Clarion Process template is not suitable for record processing when using a SQL Driver - every single record is processed one at a time. The script below is a faster way of updating all the rows in the table.

```

UPDATE ZIPCODES
SET ID_CITY = c.ID_CITY FROM dbo.CITY c
WHERE c.CITY_NAME = CITY
GO

```

You can get the script containing all these steps from the downloadable zip at the end of this article.

## Clarion dictionary changes

It is necessary to make a corresponding change to the `CITY` field in the Clarion dictionary (`INV_SQL.DCT`). You have to change the size from 26 characters to 31 characters; if you don't do this, you will get an error when you attempt to open the `ZIPCODES` table. Later, I will eliminate fields like `dbo.ZIPCODES.CITY`, `dbo.ZIPCODES.STATE` and correct references to them in the application.

## Summary

In this article, you will notice that all conversions and data manipulation have been done in SQL, because SQL is more efficient in manipulating large data sets than standard Clarion

code. You may want to stick to your traditional way of data conversion, that is, using the Clarion language. However, getting comfortable with SQL will greatly improve your efficiency and give you a greater edge. I remember a [poll](#) taken a while ago by ClarionMag on developers who intend to convert their applications to SQL. I believe figures from those results would have doubled by now. It is nice having Clarion do the work of interfacing and data manipulation, but to get the most out of SQL databases you need to go beyond Clarion code and improve your SQL skills.

[Download the source](#)

---

*[Ayo Ogundahunsi](#) presently lives in Henderson, Nevada, about ten minutes from Las Vegas. He works for [Impac Medical Systems Inc.](#), the leading company in cancer therapy software (written in Clarion). Impac has its headquarters in Mountain View, California. Ayo is married to Ayodola, and they have two boys, Darren and Joshua.*

## 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.

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

[Advanced Search](#)

[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)

**Related Categories**

[Home](#)

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

[Marketing](#)(1)

[Back Issues](#)

[COL Archives](#)

## The Program's Finished. Now What?

**Related articles**

[Advertise](#)

by **Braxton DeGarmo**

[Reader](#)

Published 2002-03-13 [Printer-friendly version](#)

[Comments](#)

[Call for](#)

[Articles](#)

[Log In](#)

[Membership/ Subscriptions](#)

Okay, you've finished the greatest piece of software since, uh, Clarion? It's a program that millions of users will want to install on their systems, and at only \$19.95, have you got a deal for them. There's only one drawback: How will they learn about this fabulous product?

[FAQ](#)

Marketing. Huh? I'm a programmer, not a marketer. And my CPA says I shouldn't plan on hiring a marketing staff until I've sold at least 5,000 units. Of course, if you choose to market the program, you soon realize you also inherit the task of selling and the headaches of support. For some, this is not their ultimate dream. If you want to stick to development and let others do the marketing, selling, and support, you have two options. The first choice is to sell the program to a third party. They assume all of the risks and expenses of selling the product, while you earn a nice return on your work. But there's a Catch-22 here. You're selling the program because you don't want to market it. Yet, without prior marketing, the product has no track record and your sale price could be a fraction of the potential worth of the program. You might get nothing more than what you could earn as a contract programmer, while the new owner of the software rakes in the anticipated millions. Not to mention that you've *sold* the program and have no control over it. Period.

[Privacy](#)

[Policy](#)

[Contact Us](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)



So how do you avoid the risks while retaining your rights and

reaping the just rewards of an ever-growing market and its resultant sales? That's option two, licensing.

## Licensing

Licensing is the granting of specific rights to a third party to do whatever those rights allow them to do with your program. In the scenario we're discussing, that means all of the rights to market, sell, and support the program. In other words, to act as the publisher of the program. This can be done on an exclusive or non-exclusive basis.

"Exclusive rights" gives that company, and *only* that company, the rights spelled out in your agreement. For the great majority of software, this is what a third-party publisher will expect. And what you should plan on granting them. After all, they're taking the financial risks to market, sell, and support your creation, and they deserve a return on that investment. Within the Clarion community, Michael Brooks' "ClarioNet" is a great example of exclusive licensing.

The role of non-exclusive rights might come into play if you have a program that a publisher wants modified or customized to their specific needs. They get the rights to publish that customized version, while you retain the rights to do the same for other companies or to sell a version of the program on your own. Friedrich Linder's "SetupBuilder" and "SetupBuilder for Clarion" are a good example of non-exclusive licensing.

Once you've found a suitable candidate who is interested in publishing your program, and you've agreed to the level of rights to be granted, what's next? Most publishers will have a standard licensing agreement, particularly if they're established and successful. The more successful they are, the more they will hint, or downright insist, that this is the only agreement they use. Don't be fooled by that. All agreements can be modified. Besides, if they're so successful, and they seriously want your program, then they know they have a potential profit-maker, and that gives you the upper hand.

I learned this lesson firsthand several years ago when I was approached by the nation's largest medical publisher. A friend and I had developed the very first PC-based drug

interaction database in the mid-1980s and had successfully marketed it despite growing competition from multi-million dollar companies. Still, after ten years the product cycle was waning, as was our interest in continuing as the sole marketer of the program. So, the timing was right when this publisher called and said they wanted to license our database and software. In the end, our contract looked nothing like their standard contract.

We gave them exclusive publishing rights, while retaining all ownership and copyrights. We continued to maintain and build the database. In return, we received a royalty every quarter. After five years of that, we finally grew tired of being editors, became frustrated with a company that was under its third owner in four years, and negotiated the sale of the database to them.

This second agreement was by far the more difficult of the two. There were several reasons for that, but the main factor was that the first agreement was not nearly as good as we had believed when we signed it. So, from the perspective of perfect 20/20 hindsight, as I move into "What's next?" I'll also point out the pitfalls I wish we could have avoided.

Most licensing agreements will start with the usual definitions. The product being licensed, as well as the rights being granted, will be defined. While this is pretty straightforward, make sure you read this carefully. On the first draft of our agreement, the wording in this section required us to provide them data in a format we were not equipped for without additional hardware and many additional hours of time each quarter. When reminded that we were two men to their staff of hundreds, they quickly acquiesced and removed that clause.

## **Royalties**

The first major stumbling block in a licensing agreement is almost always the royalty. After all, you want as much money as you can squeeze out of them, and they want to stop being squeezed. Coming from a book-publishing heritage, most early software publishers used models that called for royalties between 5% and 15% of net receipts -- the monies received after prepaid freight, postage, handling,



insurance, and tax expenses. Out of these net receipts, they also paid the expenses for production, packaging, marketing, sales, and support. Plus, they had to provide discounts to their distributors and retailers, while ultimately making a profit to stay in business. All of this was/is true. They have those expenses, but software is considerably cheaper to produce than books.

In today's market, a new wrinkle makes it even more important that you scrutinize how they plan to sell your program. If they're going to place it on a CD and in a box for retail shelves, they will incur a higher expense. But if they plan on Internet downloads as a significant percentage of their sales, they greatly reduce, if not nearly eliminate, the expenses of production, packaging, and distributor discounts... and you, as the author, should also benefit from the improved profit margin.

The Internet also directly affects another aspect of royalties. From the old book-publishing model, overseas or international sales were always rewarded a lower royalty than national sales due to increased costs of international business. Our agreement's first draft even differentiated rates for Canada and Mexico at a time when free trade agreements were opening these borders. We ultimately settled on one rate for North American sales and an international rate elsewhere. But today the Internet dissolves these borders, so be prepared to negotiate should they offer you a lower royalty for international sales.

The final aspect of royalties that typically creates some tension is that of advanced royalties. This is money paid up front, typically in anticipation of future royalties. That means you won't see another check until you have accrued royalties in excess of that first payment. Advance royalties create a strain because authors want them and publishers don't want to give them. The publisher simply does not want to pay out money before seeing sales materialize. They might even ask to make the advance refundable, which means that if sales within a given period do not support the advance, you have to give back the difference. As an author, hold out for a non-refundable advance, using a smaller amount as a compromise if necessary. This adds incentive to the publisher's doing his job.

## Copyrights

When it comes to copyrights, my advice is simple. Make sure you retain all copyrights. Some publishers will try to claim copyrights in an agreement. If you let them, and somewhere down the road you part company with them, their ownership of the copyrights will give you a major headache.

## Term and options

Even if you never foresee publishing your program yourself, make sure your agreement has a set term, or time period. When we entered into our first agreement, we were tired of being publishers and saw no reason why we would ever take back those rights from the nation's number one publisher in its field. By the time this company was up for sale for the third time in as many years, we were regretting the term of our agreement. The language defining our term was vague, and in hindsight, was to the advantage of neither party. But when it came time to renegotiate, this ambiguity hurt both parties. Be specific and plan on a shorter term (1 to 5 years) rather than a longer term. Automatic renewals are fine, but you need the ability to get out of the agreement if conditions make that a necessity.

Along with the time period of the agreement, make sure any options to purchase are also specifically spelled out. This means placing a clause in the agreement that gives the publisher the first right to buy your program should you decide to sell it. Giving the publisher this option is a courtesy, but one you should seriously offer. The pitfall to avoid here is basing your selling price on sales targets that might never be reached. Make sure you also consider a worst-case scenario. After all, your program has yet to prove itself.

Although our agreement spelled out a formula for selling the program, we could never have foreseen the company itself going on the sale block three times. They were so busy catering to new owners, product sales suffered. For a while, one new owner had the entire electronic publishing division on hold while trying to decide whether to even continue in that end of the business. When we decided to sell the program, the formula that once looked so rosy was almost

worthless. Fortunately, our contact person within the company also recognized that the problem was due largely to their failure to perform and was willing to negotiate a new purchase agreement. (But not until after several heated meetings and several cases of antacids.) Had we spelled out a minimum sale price in the first agreement, we could have avoided the unpleasantness.

Licensing can be a rewarding way to reap the financial rewards of your programming. Sure, you see only a fraction of the profit, but you avoid the hassles and eliminate the financial risks of publishing. In the midst of negotiations like this everyone is upbeat and the future is bright. And that's how it should be. But the wise developer will always count the costs and plan for contingencies. The key to a successful licensing agreement is to make sure you review it thoroughly and consider the worst that might happen. Then you won't be surprised or disappointed.

---

*[Braxton DeGarmo, MD](#) has been programming with Clarion since 1989. As a practicing physician, he needed the most efficient development tool possible to continue his medical programming, and Clarion was the perfect prescription. In the past two years, he has broadened his programming to several vertical markets, including sportfishing ([www.cyberanglers.com](http://www.cyberanglers.com)) and writing fiction ([www.authorsassistant.com](http://www.authorsassistant.com)). Braxton has completed several novels but has yet to be published.*

## Reader Comments

[Add a comment](#)

**[Braxton, Great article. The sharing of this kind 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 ...



[Home](#)   [COL Archives](#)

[Advanced Search](#)

[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)

**Related Categories**

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

[Topics](#)

**Related articles**

## Creating A SCADA Interface With Clarion (Part 1)

by Jay Guengerich

[Advertise](#)  
[Reader Comments](#)

Published 2002-03-14   [Printer-friendly version](#)

[Call for Articles](#)  
[Log In](#)  
[Membership/ Subscriptions](#)  
[FAQ](#)  
[Privacy Policy](#)  
[Contact Us](#)  
[PDFs](#)  
[Freebies](#)  
[Open Source](#)

There are many industries that rely on automated process controls for creating a product or providing a service. Supervisory Control and Data Acquisition (SCADA) is the term often used to describe an industrial automation system. Sometimes all of that acquired data needs to be stored somewhere for future reference; since Clarion is very good at storing, manipulating, and viewing data, a Clarion application is a good fit for a SCADA system data storage interface. Some features that are especially useful are the `OVER` declaration, the `EVALUATE` function, string slicing, and bitwise functions.

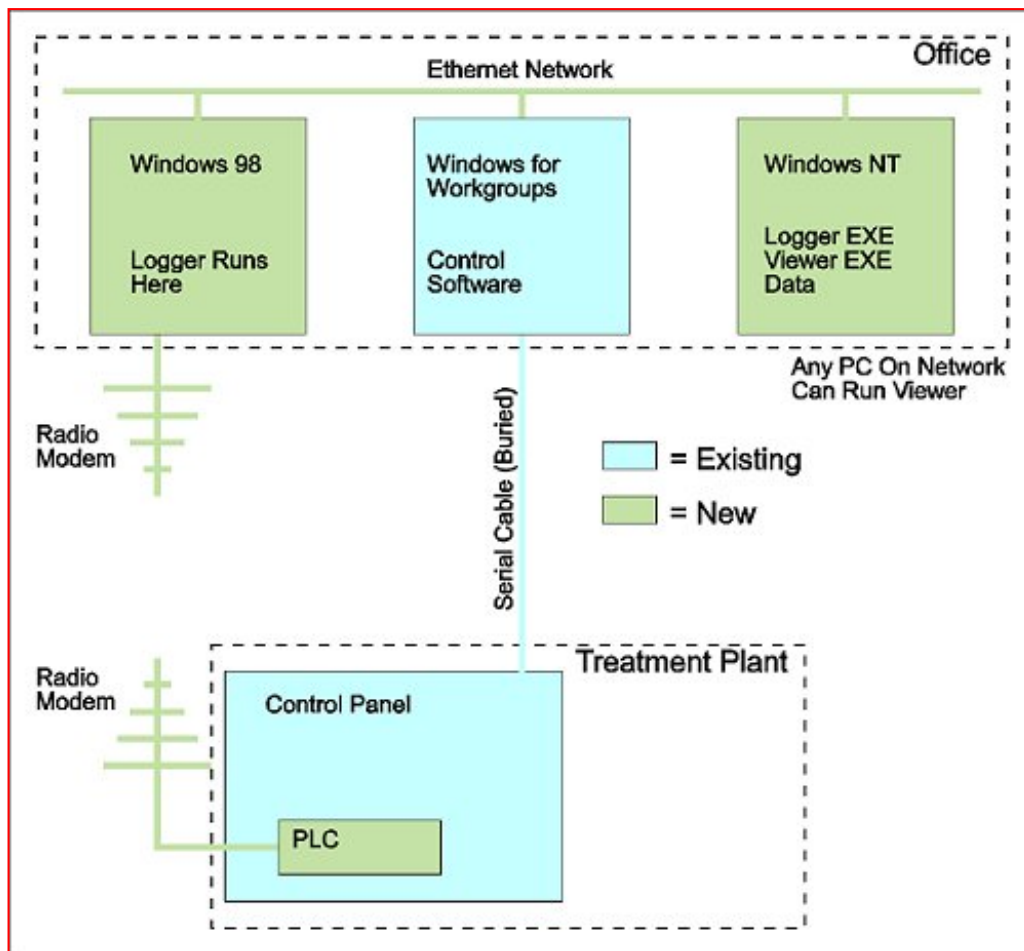
One of the companies I work for operates a wastewater treatment plant. The current SCADA system (see Figure 1) consists of proprietary software on a PC communicating with digital and analog I/O modules connected to various equipment. The PC is a 486-33 running Windows for Workgroups 3.11 located in the office building. An RS-485 serial cable runs to the control panel housing the I/O modules in the treatment plant several hundred feet away. This control panel also contains timers, fuses, relays, switches, and instruments, and the operator can control the system manually from the panel if necessary. This system was OK when it was installed six or seven years ago, but every time there is a significant rain it has to be run manually to make sure the treated discharge meets newer, more stringent EPA permit requirements.

A better control system would react more quickly to handle



varying conditions such as high flow. This is especially important at night when no one is on-site. It would also provide better treatment under all conditions to meet future EPA requirements.

The new control system will eventually replace the PC and IO boards with a Programmable Logic Controller (PLC). The first goal was a "proof of concept" accomplished by installing the PLC and sensing the same inputs as the PC without disturbing the existing control system. Figure 1 shows how the new system and the original system coexist. An [AutomationDirect DirectLogic 405 series PLC](#) was installed. A small Clarion program running on a networked PC acts as a data logger using a serial connection to the PLC. The physical link to the PLC is actually an [Apex Wireless SS-200 900 MHz wireless modem pair](#) (burying 400-500 feet of cable, with several corners required, wasn't cost effective). A second Clarion program allows users to view and analyze the data. The logger and viewer executables and the data files are all stored on the server.

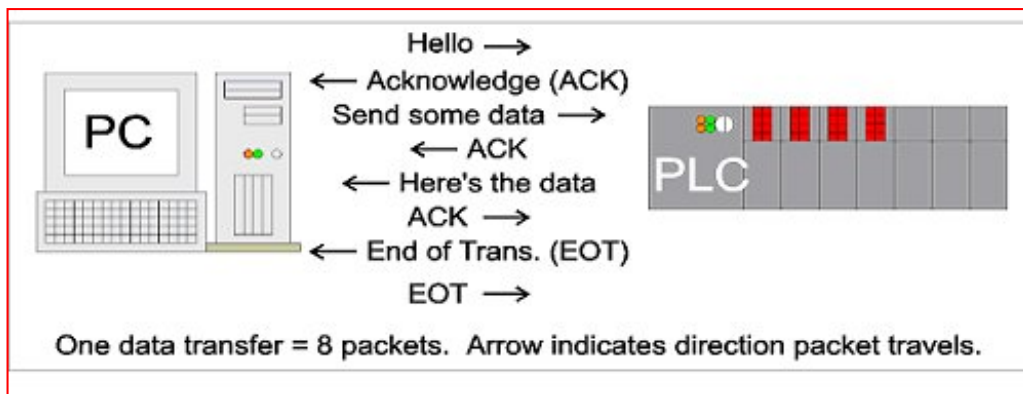


**Figure 1. System Components**

There are several advantages to the new design. First, the automated treatment process (eventually) won't depend on a Windows PC. Second, the data logger and viewer/analyzer software can be generic enough to work with other SCADA systems. Third, the control system can be expanded to include additional equipment. Finally, the control algorithms can be changed if necessary.

## Data logger communication protocol

The data logger program "talks" to the PLC using Automationdirect's *DirectNET* protocol over the serial link. I use [WinEvent](#) for serial port access, but any serial port functions would work. A typical conversation using DirectNet looks like this:



**Figure 2. DirectNET Conversation**

Because this conversation includes significant overhead (including a checksum), all of the data is transferred in one block. The PLC software copies all of the I/O points and variables to a contiguous block of memory starting at a specific location. The Clarion program requests the proper amount of data starting at that location. A data request is made every 15 seconds using a window with the `TIMER(1500)` attribute.

## Data storage

Data is stored in a file on the network with a time and date stamp. The data logger application therefore has two files so far, `ranges` and `plclog`:

```
ranges FILE, DRIVER( 'TOPSPEED' ), PRE( RAN ), CREATE, BINDABLE, THREAD
ByRangeStart KEY( RAN:RangeStart ), NOCASE, OPT, PRIMARY
Record RECORD, PRE( )
RangeStart STRING( 5 )
ByteCount USHORT
```

```

                END
            END
    plclog FILE, DRIVER( 'TOPSPEED' ), PRE( PLC ), CREATE, BINDABLE, THREAD
    byRecID KEY( PLC:RecID ), NOCASE, OPT, PRIMARY
    byDateTime KEY( PLC:LogDate, PLC:LogTime ), DUP, NOCASE, OPT
    PLCData MEMO( 255 ), BINARY
    Record RECORD, PRE( )
    RecID LONG
    LogDate LONG
    LogTime LONG
    BytesUsed LONG
        END
    END

```

RangeStart is a STRING(5) because the PLC uses a five digit octal addressing scheme. However, the address must be converted to a four byte Hex string and incremented by one for the DirectNet protocol (don't ask me, I didn't design the protocol!). The conversion is fairly simple (the EVALUATE function really comes in handy sometimes!):

```

! Convert octal string to long integer and add 1
LongEquiv = EVALUATE(RAN:RangeStart & 'o') + 1
! Convert long integer to hex string
HexAddr = HexDigits[BSHIFT(BAND(LongEquiv, 0F000h), -12)+1] |
          & HexDigits[BSHIFT(BAND(LongEquiv, 0F00h), -8)+1] |
          & HexDigits[BSHIFT(BAND(LongEquiv, 0F0h), -4)+1] |
          & HexDigits[BAND(LongEquiv, 0Fh)+1]

```

LongEquiv is a LONG, HexAddr is a STRING(4), and HexDigits is a STRING(16) containing '0123456789ABCDEF'. Note that HexDigits[LongEquiv/1000h + 1] would give the same result for the first character.

Storing BytesUsed in the log file allows the length of the data block to increase without worrying about accessing unused portions of the memo field in older records. The viewer/analyzer software must not try to access data offset by more than BytesUsed.

## Application behavior

The application starts with its window hidden (0{PROP:Hide} = TRUE) and places an icon in the system tray using WinEvent's Add Icon to the InTray extension. The app uses WinEvent's AutoShutdown extension to terminate if Windows shuts down. The following code in the Taskbar Icon Message Processing – Mouse Double Left embed (added by WinEvent) activates the main window when the user double-clicks the system tray icon:

```
0{PROP:Hide} = FALSE
```

```
WinBringToFront()
```

When the user hits the Escape key, the following code in the Local Objects, ThisWindow TakeCloseEvent embed, after Parent Call, hides the window again:

```
IF ForceClose <> TRUE THEN      ! don't really exit app.
    0{PROP:Hide} = TRUE         ! hide window
    RETURN LEVEL:Notify        ! "shortcut" template code
END
```

To actually exit the application, the user clicks an Exit button based on the CloseButton control template. The following code, in the Control Events > ?ExitButton > Accepted embed, before Generated Code, gives the user three choices:

```
HoldMess = MESSAGE('Press Hide to let ' |
    & 'PLC Logger continue in the background.' |
    & '|Press Exit to stop the PLC Logger.' |
    & '|Press Cancel to return to the main window.' |
    , 'Options', ICON:Question, 'Hide|Exit|Cancel')
CASE HoldMess
    OF 1      ! Hide
        ForceClose = FALSE
    OF 2      ! Exit
        ForceClose = TRUE
    OF 3      ! Cancel
        RETURN LEVEL:Benign ! "shortcut" template code
END
```

In the first two cases, the TakeCloseEvent code listed earlier executes to hide or close the window.

## Data viewer

Now that there is a data logger to store all of the information on the network, the plant operators and managers should be able to view and analyze the data. The data viewer application takes care of that task.

## Data dictionary

The data block transferred from the PLC contains three types of variables (the PLC has other data types, but they are not used): Binary (On/Off I/O points), four digit unsigned packed decimal (decimal position not stored), and four byte floating point (formatted like Clarion's SREAL type). To interpret the data correctly, the viewer application needs a file that describes what the data block contains. Here is the layout I use:

```
elements FILE,DRIVER('TOPSPEED'),PRE(ELE),CREATE,BINDABLE,THREAD
```



```

byElementID      KEY(ELE:ElementID),NOCASE,OPT,PRIMARY
byWordBit        KEY(ELE:Word,ELE:Bit),DUP,NOCASE,OPT
byDescription    KEY(ELE:Description),NOCASE,OPT
byGroupID        KEY(ELE:GroupID),DUP,NOCASE,OPT
Record           RECORD,PRE( )
ElementID        LONG           ! unique ID for each variable
Description      STRING(100)    ! text description of variable
Word             USHORT          ! which word contains variable
Bit              USHORT          ! if binary, which bit of word
RealNumber       STRING(1)      ! 'Y' if SREAL, 'N' otherwise
Multiplier       DECIMAL(7,3)   ! Multiplier for decimal vars
RoundTo          DECIMAL(4,3)   ! Rounding for display purposes
Units            STRING(20)     ! units (feet, minutes, etc.)
ReversedOnGraph  STRING(1)     ! 'Y' if reverse logic (1=off)
StatusText       STRING(20)     ! Text if 0|Text if 1
GroupID          LONG           ! group ID for text display
                END
            END
    END

```

Since each PLC address contains a 16-bit word, I use a word offset instead of a byte offset. The following are examples of the various kinds of data followed by a description of each one.

ElementID	6	10	11	20
Word	2	7	8	15
Bit	3	0	0	0
Real Number	N	N	N	Y
Multiplier	1	0.01	1	1
RoundTo	0	0.1	0	0.01
ReversedOnGraph	Y	N	N	N
StatusText	Open  Closed			
GroupID	0	0	1	0

**ElementID 6:** The third bit of the second word in the data block is a reverse-logic binary value. The status will be shown as "Open" if the value is 0 or "Closed" if the value is 1.

**ElementID 10:** The seventh word in the data block is a BCD number with two decimal places (determined by Multiplier). It will be rounded to one decimal place for display (determined by RoundTo).

**ElementID 11:** The eighth word in the data block is a BCD

number with no decimal places. The value will be used to select the proper text description from the group with a GroupID of 1 (see below).

**ElementID 20:** The 15<sup>th</sup> and 16<sup>th</sup> words (four bytes = two words) in the data block contain a short real number. The value will be rounded to one decimal place when it is displayed.

Some of the decimal variables contain a sequence number that needs to be translated to a text phrase, as shown by ElementID 11 above. This is done using a parent-child relationship with a pair of files to handle more than one type of process:

```

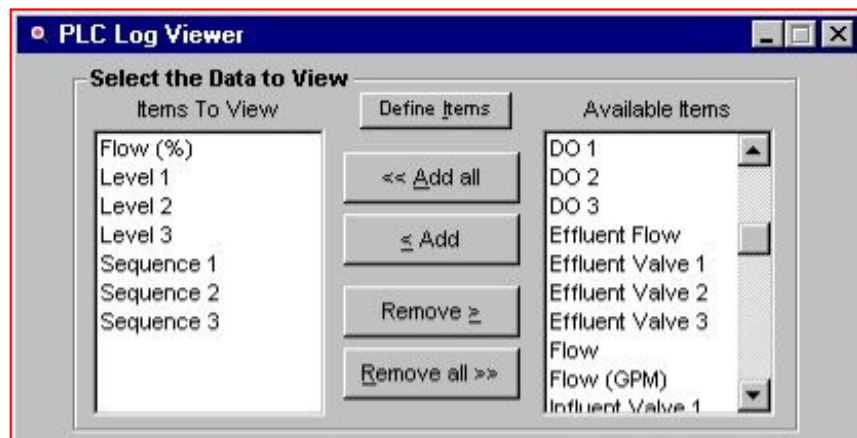
groups      FILE,DRIVER( 'TOPSPEED' ),PRE(GRO),CREATE,BINDABLE,THREAD
byGroupID   KEY(GRO:GroupID),NOCASE,PRIMARY
byGroupName KEY(GRO:GroupName),NOCASE
Record      RECORD,PRE( )
GroupID     LONG           ! group ID for text display
GroupName   STRING(50) ! name of this group
            END
            END
messages   FILE,DRIVER( 'TOPSPEED' ),PRE(MES),CREATE,BINDABLE,THREAD
byGroupMessage KEY(MES:GroupID,MES:MessageID),NOCASE,OPT,PRIMARY
Record      RECORD,PRE( )
GroupID     LONG           ! group ID for text display
MessageID   LONG           ! if variable contains this...
Message     STRING(50) ! ...display this text
            END
            END
            END

```

Going back to the previous example, if ElementID 11 (the eighth word) contains 6, the application looks for a record in the messages file that has GroupID = 1 and MessageID = 6. The text in the Message field is the information to display.

## Status list selection

The user can choose which variables to display on the status screen. The application uses a dual list interface as shown below. The Define Items button opens a standard EIP screen for filling in the elements file. That screen includes a button that opens another EIP screen for the groups and messages files.



**Figure 3. Dual List for Choosing Variables**

To move an item from one list to the other, the user can double-click on the item or select the item and click the Add or Remove buttons. The two lists are list box controls, not browse control templates. The Items To View list is from a queue called ChosenQ and the Available Items list is from a queue called AllQ. Both queues have the same fields as shown below:

```
AllQ          QUEUE,PRE(AQ) ! same fields as elements file
ElementID    LONG
Description   STRING(100)
Word         USHORT
Bit          USHORT
RealNumber   STRING(1)
Multiplier   DECIMAL(7,3)
RoundTo      DECIMAL(4,3)
Units        STRING(20)
ReversedOnGraph STRING(1)
StatusText   STRING(20)
GroupID      LONG
            END
```

When the window opens, AllQ is filled from the elements file:

```
FREE(AllQ)
SET(elements)
LOOP UNTIL Access:elements.Next() <> LEVEL:Benign
    AllQ :=: ELE:Record
    ADD(AllQ)
END
SORT(AllQ, AllQ.Description)
FREE(ChosenQ)
```

Both lists have MouseLeft2 set as alert keys. Code in the AlertKey embed for each list posts an EVENT:Accepted to the Add or Remove button.

The code for the Add All button moves all of the Available Items into ChosenQ:

```

IF RECORDS(AllQ) > 0 THEN
  GET(AllQ, 1)
  LOOP UNTIL ERRORCODE()
    ChosenQ :=: AllQ
    ADD(ChosenQ)
    DELETE(AllQ)
    GET(AllQ, 1)
  END
  SORT(ChosenQ, ChosenQ.Description)
  DISPLAY
END

```

The code for the Add One button moves the currently selected item from the Available Items list into ChosenQ:

```

IF RECORDS(AllQ) > 0 THEN
  IF CHOICE(?AllList) > 0 THEN
    GET(AllQ, CHOICE(?AllList))
    IF NOT ERRORCODE() THEN
      ChosenQ :=: AllQ
      ADD(ChosenQ)
      SORT(ChosenQ, ChosenQ.Description)
      DELETE(AllQ)
      DISPLAY
    END
  END
END

```

The code for the RemoveOne button moves the currently selected item from the Items To View list into AllQ:

```

IF RECORDS(ChosenQ) > 0 THEN
  IF CHOICE(?ChartList) > 0 THEN
    GET(ChosenQ, CHOICE(?ChartList))
    IF NOT ERRORCODE() THEN
      AllQ :=: ChosenQ
      ADD(AllQ)
      SORT(AllQ, AllQ.Description)
      DELETE(ChosenQ)
      DISPLAY
    END
  END
END

```

The code for the Remove All button moves all of the Items to View into AllQ:

```

IF RECORDS(ChosenQ) > 0 THEN
  GET(ChosenQ, 1)
  LOOP UNTIL ERRORCODE()
    AllQ :=: ChosenQ
    ADD(AllQ)
    DELETE(ChosenQ)
    GET(ChosenQ, 1)
  END
  SORT(AllQ, AllQ.Description)
  DISPLAY
END

```

Setting the list box format takes two steps. First, choose the Description field from the appropriate queue in the List Box Formatter. Then add "#2#" to the `FORMAT` attribute in the window editor (the ... button next to the Window button on the procedure properties screen) or code a `{PROPLIST:FieldNo, 1} = 2` for each list (pick an embed after the window opens). The second step tells Clarion that the Description field is the second field in the queue.

It's just about time to display the information to the user! Next week I'll describe the status display and the related code.

---

*[Jay Guengerich](#) graduated from Valparaiso University (Valparaiso, Indiana) in 1993 with a BS in Computer Engineering. He chose engineering because he didn't want to "spend all day sitting at a desk writing code." After graduation, he passed the Engineer in Training exam, and was then hired by Reliable Development Corp. where he has spent quite a bit of time sitting at a desk writing code. Before his boss discovered Clarion for Windows (shortly before version 1.5 was released), Jay mostly programmed in FoxPro and VBA. In the last several years he has begun working with single board computers, PICs, embedded systems, and industrial control projects. He still pretends he is an engineer by keeping his membership in IEEE and the IEEE Computer Society current. He is also a member of Tau Beta Pi, the Engineering Honor Society.*

## Reader Comments

[Add a comment](#)

**[A note on the ElementID table: Normally fields in a file...](#)**

**[Yeah! This article is one of the best I've seen for a long...](#)**

**[RE: Code for download. I'll have to talk with my boss...](#)**

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.



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

[Advanced Search](#)

[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)

### Related Categories

[Home](#)

[Topics](#) > [Forms](#) > [Forms - using](#)

[Forms - using](#)(11)

[Back Issues](#)

[COL Archives](#)

## Secondary Forms (Part 1)

[Advertise](#)

by **Steven Parker**

[Reader](#)

Published 2002-03-15 [Printer-friendly version](#)

[Comments](#)

[Call for](#)

[Articles](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy](#)

[Policy](#)

[Contact Us](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

Sometimes there are too many fields to place on a single window. Sometimes fields can be grouped according to a logical division of data types. Sometimes, spreading fields so that not all of them are displayed simultaneously makes the data easier to digest. Sometimes the user wants multiple screens and sometimes it just looks better to use multiple pages to display data.

A very common requirement is for a single window to show the majority of the data while less frequently viewed, less critical data is on another page. The Clarion templates (ABC and Legacy) provide two built in-ways to spread entry fields across multiple (virtual) windows: tabs and wizards. But, many Clarion developers do not like these alternatives for design or aesthetic reasons, so they use windows for these supplementary data displays.

In many cases, the use of a window for additional pages of information is a "legacy" issue. That is, it's the legacy way of doing things ("that's the way our previous program worked," "that's what we're used to" and the like). In some cases, the Clarion application was created before TopSpeed cloned the Windows tab control (in Clarion for Window 1.5, I believe).

Since additional windows are usually called via buttons, I suppose it is more accurate to say that Clarion supports three ways of creating multiple page forms: tabs, wizards and

### Related articles

[Handling Multiple Update Forms](#)

[Implementing Read-Only Checkboxes](#)

[Checkboxes For Many-to-Many Relationships](#)

[Checkboxes For Many-to-Many Relationships: The Source Code](#)

[Secondary Forms \(Part 2\)](#)

[Better Ellipsis Buttons](#)

[Tomato Cans - Word-Wrapping Strings, and You](#)

[Calling Form Procedures](#)

[Making Entry](#)



buttons. Indeed, at various places in the Clarion environment, each of the three methods is used.

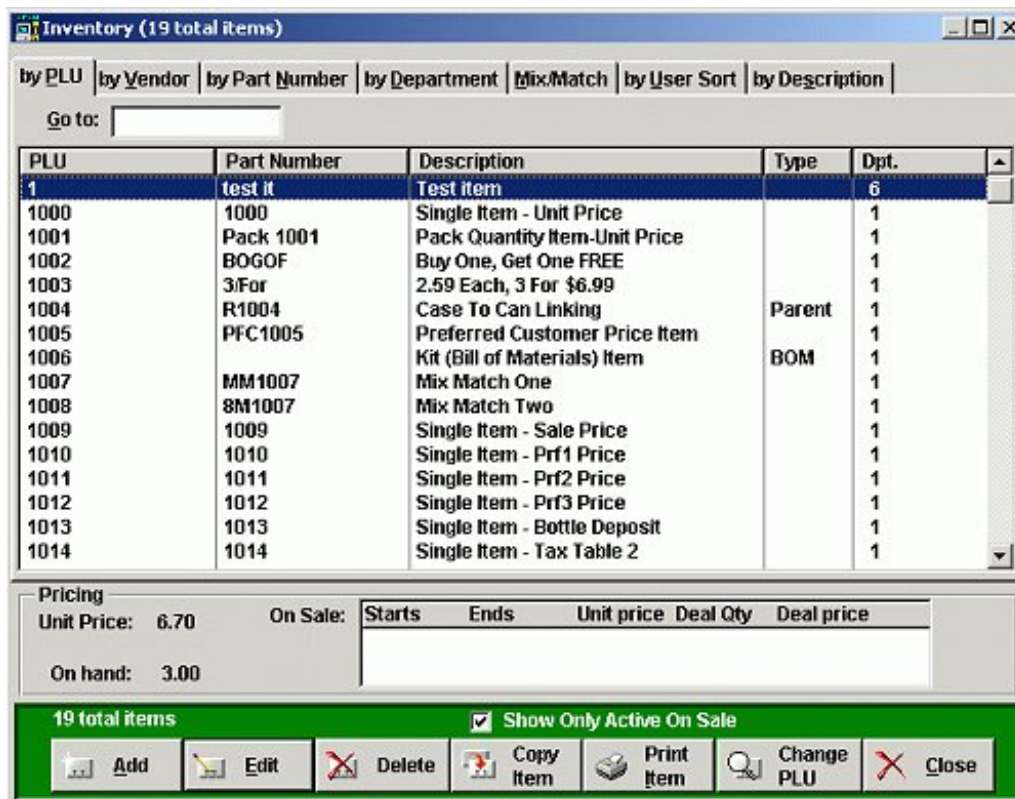
[Locators recognize EnterKey](#)

Having said this, it is important to say that not all of these methods are equally easy to implement, at least so that you get the exact results desired/expected.

[Mike's Meanderings - Multi-Threaded Browsers and Forms](#)

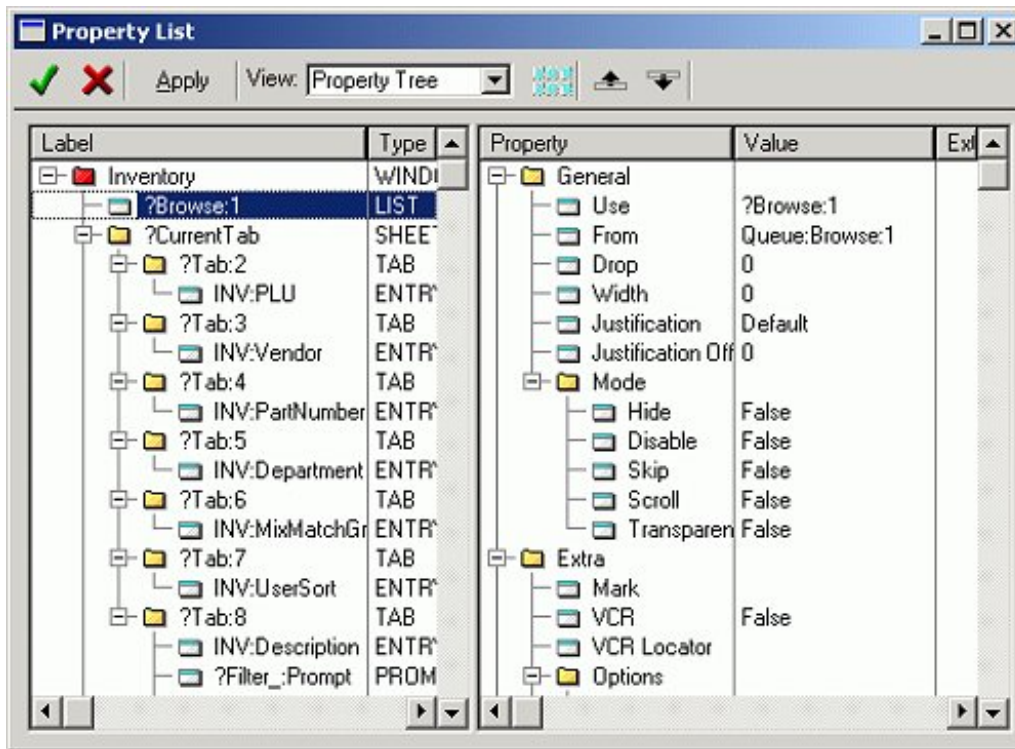
## Tabs

Tabs are usually associated with browses and providing users with one-click access to multiple sort orders, as in Figure 1. (Actually, the typical reference to "tabs" is technically incorrect. What we call a "tab" is actually a Sheet control which may contain additional tab controls. Oh, heck, you know what I mean.)



**Figure 1. Typical use of tabs**

When used as shown in Figure 1, the tabs are, in fact, empty. The browse box is populated before the Sheet control, which contains the tabs. The browse is not logically *on* the sheet (i.e., not inside the `Sheet` structure). Populating the browse and tabs this way ensures that when the window opens, the browse is the selected, active, control. See Figure 2.



**Figure 2. Normal control population order**

If individual tabs contain a control, it is most likely to be a locator for the browse order associated with the tab (as shown above). But, in fact, a tab is nothing but a control nested inside another control, a sheet control (sometimes called a "container" or "container control"). Pragmatically, a tab is just another page. Therefore, it can be used as a container for other controls, say, entry fields, child browses, RTF or text controls. So, on a form, tabs can contain sets of fields grouped by some common denominator:



The screenshot shows a software window titled 'Inventory Item' with several tabs: 'General', 'Options', 'Vendors', 'Descriptions', 'On Sale', and 'Sales History'. The 'General' tab is active. The form contains the following fields and values:

- PLU: 1 (This record created: 1/17/01)
- Part Number: test it
- Description: Test item
- Item Type: Regular
- Department: 6 (Dept 6)
- User Sort: [empty]
- Vendor: DEMO (Primary Demonstration Vendor)

Cost/Unit section:

- Vendor Cost: 0.002
- Last Cost: 0.002
- Standard Cost: 0.200
- Average Cost: 0.002
- Price Method: 6 (Prorated Table) [Table button]
- Unit Price: 6.70 (97.01% Margin)
- Mix / Match: [empty]
- List Price: 0.00

Inventory Status section:

- On hand: 3
- On Order: 0
- Order Point: 0
- Order Multiple: 0
- Order Pack Qty.: 1
- Date Last Sold: 2/18/02
- YTD Sold: 10
- Last Received: [empty]
- YTD Receipts: 0
- Sold Since Rec.: 10
- Last Phy. Inv.: [empty]
- Location: [empty]

At the bottom, there is a 'Print' button, a status bar showing 'PLU: 1', and 'Ok' and 'Cancel' buttons.

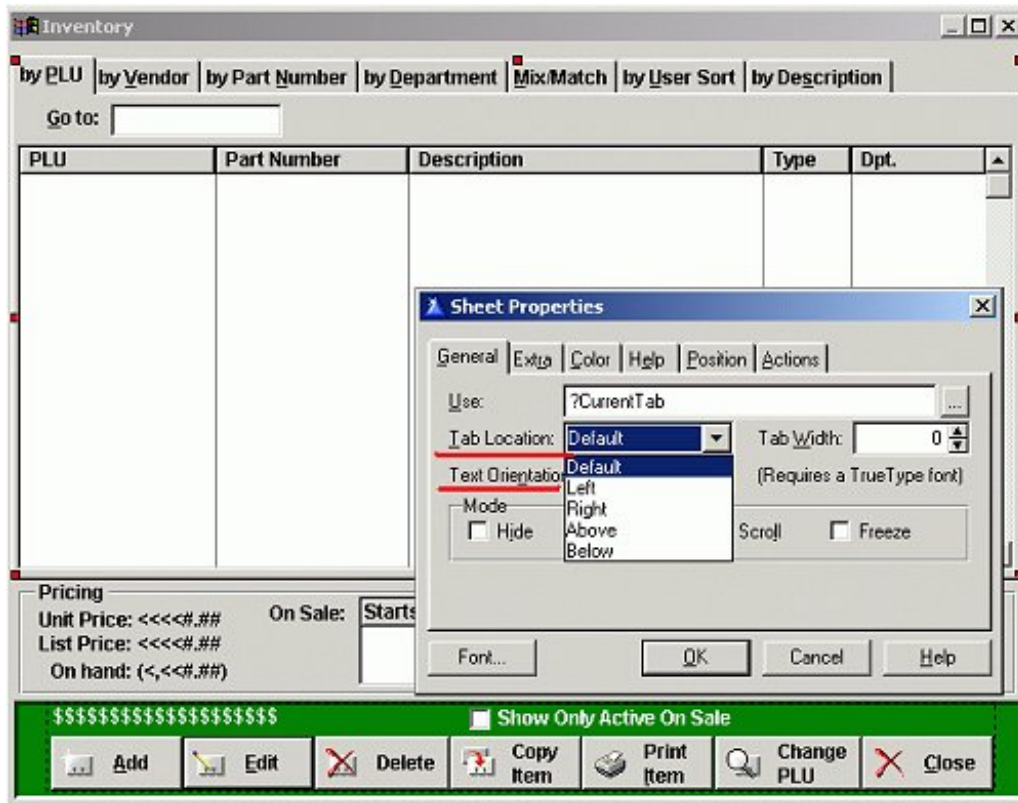
**Figure 3. Tabs on a Form template procedure**

Creating a tabbed data entry form is quite easy. I usually start a new form using the Form Template wizard. The Form wizard provides a sheet, even if there are not enough entry fields to require a second tab/page (see the demo app). If more tabs are required, it is a simple matter of selecting Control | Tab and clicking on the sheet control to populate a new tab.

Usually, the Form wizard populates several tabs. I resize the window and rearrange things to taste.

Starting with a window created by the Form template or the Window template (with [Save Button control template](#)), populate a sheet: Control | Property Sheet. This populates a Sheet control, the container for Tabs. It also populates the first tab. You can populate additional tabs with Control | Tab.

Sheet controls offer a number of aesthetic options. You can place the Tabs on any side of the Sheet and you can rotate the text. See Figure 4. While not as pretty as the SAXTab OCX that was so popular a few years ago, Clarion sheets/tabs not only provide a functional data entry form, they are not as difficult to populate and control as an OCX.



**Figure 4. Sheet Tab/text options**

Tabs have one very, very important advantage over multi-window forms. Because only one window is used, there is only one `Accept` loop. This means that all file I/O occurs in a single place. Buffers can't be corrupted, at least not without a bit of work on your part or a modicum of SWB (Standard Windows (mis-)Behavior).

## Wizards

Wizards, I think, are much less useful for general data entry purposes than sheets with visible tabs. Not only that, they are much more difficult to implement and they create design-time difficulties.

Because a wizard leads a user, prompt by prompt, through the data entry process, I can see how they can make the process of entering data easier for computer-phobes and -illiterates. When updating a record, however, forcing a user through a series of screens just to get to the field to be changed could easily, and understandably, cause significant ill will. Competent computer users will certainly consider imposing dire bodily harm on the developer.

Implementation of wizards isn't exactly point-and-click easy

either. To support the Back and Next buttons of a wizard, I need to know the number of tabs on the form and I have to track which tab I am on at any given time. I need two `Case` structures, one for the Back and one for the Next button, to enable/disable buttons (see the on-line FAQ, "How to Create a Wizard"). So, if a tab is ever added or deleted, a substantial amount of manual code must be maintained.

That's much too much like work.

Worst of all, once the Wizard attribute is placed on a sheet, the tabs disappear. To maintain the window design in the IDE, that attribute must be removed and reinstated. But suppose I could turn the Wizard attribute on at runtime. If I could turn the Wizard attribute on at runtime, I could put a memo, for example, on a second tab. I could create a button on the first tab and embed:

```
Select(?Tab:2)
```

in its `Accepted` embed, causing a button press to move to the tab with the memo. Then I could create a button on the second tab and make *it* select the first tab when pressed:

```
Select(?Tab:1)
```

Because the Wizard attribute is active, the tabs will not show. That is, it will appear to the user as if the buttons are toggling between two different forms (which happen to occupy exactly the same screen area). Luckily, there *is* such a runtime property I can set:

```
?CurrentTab{Prop:Wizard} = True !turn sheet into a wizard
```

If I want the line representing the sheet's border to disappear:

```
?CurrentTab{Prop:Wizard} = True !turn sheet into a wizard
?CurrentTab{Prop:NoSheet} = True !no line (sheet) showing
```

Et, voila!

Because I can place the button anywhere on the physical window, as long as it is logically on the tab, I can place both buttons in the same place and the form will appear to be a data entry form with one or more buttons to access and display other parts of the record. To see this in action, in the sample app, on the main menu, select Multiple Forms | EZ Wizard. The

best part is that, at design time, the Tabs are still visible and I can switch to/from them normally.

I think it is fairly clear that I am not a big fan of wizards. But, in this case, when I am not forced to navigate through the additional "tabs" but can use buttons or accelerator keys to access the information on them at will, I think they can be quite useful. Again, this technique has the virtue of keeping all I/O in one, controllable place.

I like *that*.

I've covered tabs and their wizardly variation and a variation on the variation. Next week I'll show how to spread a form over multiple windows.

[Download the source](#)

---

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

## Reader Comments

[Add a comment](#)

**[Coding Back/Next buttons can be made easier using the...](#)**

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.



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

[Advanced Search](#)

[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)

### Related Categories

[Home](#)

[Topics](#) > [Forms](#) > [Forms - using](#)

[Forms - using](#)(11)

[Back Issues](#)

[COL Archives](#)

## Secondary Forms (Part 2)

[Advertise](#)

by **Steven Parker**

[Reader](#)

Published 2002-03-20 [Printer-friendly version](#)

[Comments](#)

[Call for](#)

[Articles](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy](#)

[Policy](#)

[Contact Us](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

In the last [action-packed episode](#), I discussed the use of tabs and the wizard variation on tabs (actually, a variation on the typical wizard, a wizard created at runtime, which I dubbed an EZ Wizard), as ways to handle form data that won't fit on a single window. But not everyone likes tabs or wizards. A number of Clarion developers continue to use window procedures for supplementary data entry forms (not to be confused with data collection forms that might be used to prime filters for a report or process – these do not typically affect files). There are a number of reasons for using windows but a number of dangers in doing so.

The greatest danger in using a window is that you *cannot* create a second form using the Form template. Both the primary and secondary form would do file I/O. When adding a new record, the record will be added twice, once by the secondary form and once by the primary form. If there is a unique key (as there should be), the user will get the duplicate record message. When updating a record, if the user presses Ok on the second form, the changes will be saved even if the user presses Cancel on the primary form.

Not good.

To avoid this, you must create the secondary forms with the Window template which does not do I/O (at least, as long as you do not populate the Save Button control template). But this

### Related articles

[Handling Multiple Update Forms](#)

[Implementing Read-Only Checkboxes](#)

[Checkboxes For Many-to-Many Relationships](#)

[Checkboxes For Many-to-Many Relationships: The Source Code](#)

[Secondary Forms \(Part 1\)](#)

[Better Ellipsis Buttons](#)

[Tomato Cans - Word-Wrapping Strings, and You](#)

[Calling Form Procedures](#)

[Making Entry](#)



means that you must manage the record. That is, you must ensure that the record is restored to its initial state if the end user presses Cancel.

[Locators recognize EnterKey](#)

This is handled easily enough by declaring local variables that match the file variables that will be on the window. Alternately, you can use a variable, matching the record structure. In the Data embed add the following (substitute your file prefix for INV):

[Mike's Meanderings - Multi-Threaded Browsers and Forms](#)

```
SAV:Record      Like(INV:Record)
```

Now you can save the entire record when the window is opened (late in the `INIT` method, in `ABC`) and, thus, preserve its initial state:

```
SAV:Record = INV:Record
```

If the user presses Ok, nothing further is required:

**In the `Ok`, `Accepted`, embed:**

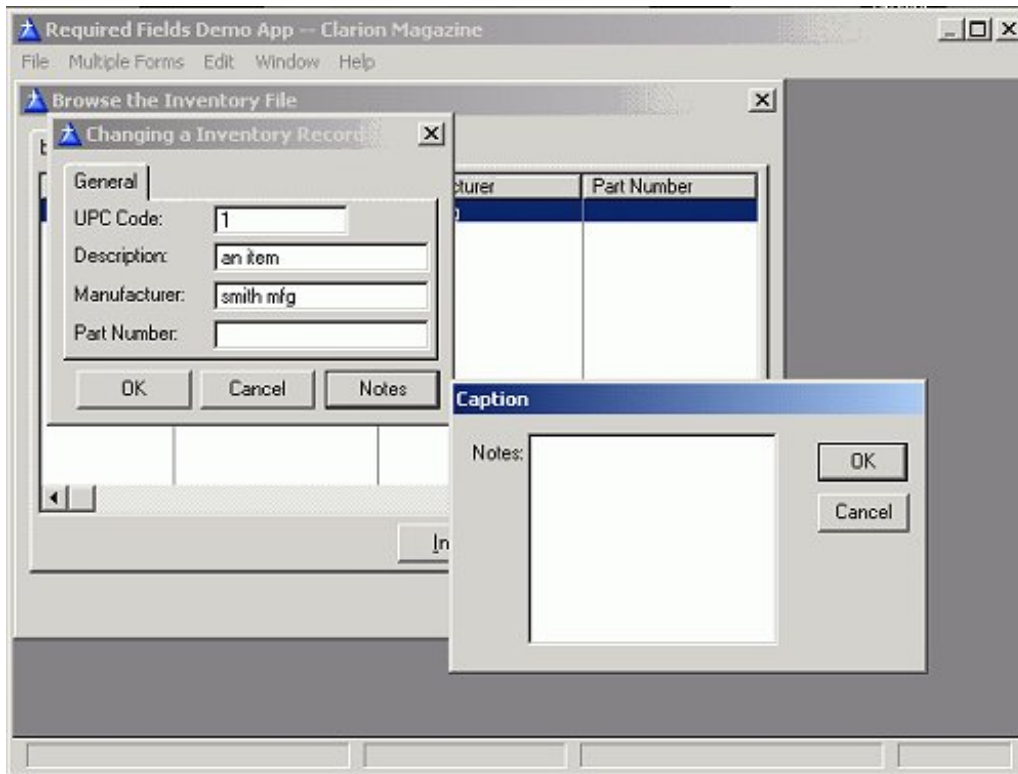
```
Post(Event:CloseWindow)
```

If the user presses Cancel, the record contents need to be rolled back; in the `Cancel Accepted` embed:

```
INV:Record = SAV:Record
Post(Event:CloseWindow)
```

Note that the OK and Cancel buttons are not created with the Save Button or Cancel templates but are just plain buttons.

Nothing could be easier. *And* the second form pops up, as expected (as shown in Figure 5).



**Figure 5. Second form using window**

Nothing could be easier. Except that it doesn't work.

In the demo app, select Multiple Forms | "Standard" Second Form; select the second form and make an entry. Press Ok to accept all changes. Return to the browse. Access the record again. Your edits are gone.

This is not quite what most of us have in mind when creating multi-page data entry forms.

## Of buffers and Bufferin

To understand why the record is not saved and how to rectify this, it is necessary to understand some fundamentals of Clarion record handling.

When a file declaration is generated in code, from a dictionary, a `Record` structure, which you did not explicitly create in the data dictionary, is added to the declaration. If you create a file definition in a data embed, this `Record` structure is required:

The `RECORD` statement declares the beginning of the data structure within the `FILE` declaration. A `RECORD` structure is required in a `FILE` declaration. Each field is

an element of the RECORD structure. The length of a RECORD structure is the sum of the length of its fields. When the label of a RECORD structure is used in an assignment statement, expression, or parameter list, it is treated as a GROUP data type.

The fact that a `Record` structure can be addressed like a `Group` is what makes the following assignment work (the default label for a `Record` structure happens to be "Record"):

```
SAV:Record = INV:Record
```

More important is the fact that the Clarion Runtime Library uses the `Record` structure:

At run time, static memory is allocated as a data buffer for the RECORD structure. The fields in the record buffer are available whether the file is open or closed.

This definition of the file buffer has been a Clarion characteristic as long as Clarion's been around. But in Windows it is no longer entirely accurate.

If the file has the `Thread` attribute, the data buffer does also; if the file is not threaded, the buffer is not. This allows threaded browses, for example, to be open against a single file (or set of files) simultaneously without the selection in one affecting the selection in the other.

Finally, consider what the Help has to say about the role of the `RECORD` buffer in updating files (note that the following does not strictly apply to SQL back ends as these are updated with `UPDATE` statements):

Records from the data file on disk are read into the data buffer with the `NEXT`, `PREVIOUS`, `GET`, or `REGET` statements. Data in the fields are processed, then written to the data file as a single `RECORD` unit by the `ADD`, `APPEND`, `PUT`, or `DELETE` statements.



Note that because browse procedures use queues (created from Views) which do not use `NEXT`, `PREVIOUS`, `GET`, or `REGET` statements, scrolling a browse does *not* normally fill the buffer. This is why fields must be declared "Hot" to display them on a browse window outside the browse box proper. "Hot" puts them in the View.

Clarion's use of the `RECORD` structure means, or should mean, that when a file is opened the first time (on a thread), a buffer is created. That buffer is owned by the thread and, once a record is read, the data in the buffer is accessible anywhere in that thread, even if the file is closed (N.B.: there is no data in the buffer until and unless a record is read).

The clear implication is that if I open a browse on a file, a buffer is or has been created (because the file is opened). When I call an update form, the data for the highlighted record *is* read into the buffer (the underlying record is retrieved). If I open another window, on the same thread, that data is available to the secondary window (whether or not the file is named in the file schematic for the procedure). The demo app accompanying this article demonstrates that this is true.

However, the implication (or at least the expectation) of this specification in the Language Reference Manual is also that if I change the contents of a field, I affect the buffer and, therefore, if a secondary window changes field contents, the first window should be able to see those changes. After all, one thread, one buffer. This, too, has been a Clarion characteristic as long as there's been a Clarion.

This is what is *not* happening; *this* expectation is not met.

It's as if the buffer is available to called procedures but is not changed by them and "passed" back to the caller. At least, this appears to be the case between the form and the window: changes in the called procedure are not visible in the caller.

Not "passing" changes in the buffer back to the caller is precisely what the `FileManager` is doing but precisely what we expect it to do.

A significant change was made in Clarion 5b's implementation of the `UseFile` method:

**UseFile** The UseFile method notifies ABC Library objects that the managed file whose opening was delayed by the LazyOpen property is about to be used. UseFile returns a value indicating whether the file is ready for use. A return value of Level:Benign indicates the file is ready; any other return value indicates a problem.

Starting in 5b, UseFile takes a single parameter, UseType:

**usetype** A numeric constant, variable, EQUATE, or expression that determines how UseFile handles the file buffer. A value of UseType:Corrupts changes the value in the file buffer but does not rely on the new value. A value of UseType:Uses changes the value of the file buffer and then uses that value. A value of UseType>Returns holds a value from the file buffer to return it to the parent. A value of UseType:Benign indicates that no special file buffer handling is requested.

UseType was introduced to allow simplified referential integrity code in the templates, switching a child record from one parent to another and recursive procedure calls (lookups within a single file, similar to the proper use of Alias files; see "[Alias - Who Was That Masked File?](#)"). Fundamentally, it addressed Clarion developers' propensity for reusing existing procedures in such a way that there was a recursive call rather than creating new lookup procedures using an Alias.

This modification forever changed our ability to cascade changes up and down a chain of windows in the straightforward way traditional for Clarion (see Dave Bayliss' "[Propitious Memory Corruption](#)" for a full discussion of the how's and why's of this change).

On first glance, it would appear that UseFile is not relevant in a secondary window. After all, the file is already opened. No one, however, has told the FileManager that the file is open, so UseFile remains relevant (sometimes, FileManager just

isn't very bright). In fact, while `UseFile` normally forces the opening of a file (i.e., is the "real" open method), it is the parameter that is important here; this parameter tells the `FileManager` what to do with changes in the buffer when the procedure exits. Thus, even if the `FileManager` knows the file state, it still needs to know how you want the buffer handled.

A careful reading of the documentation seems to indicate that the following code will allow changes in the secondary window to cascade back up to the calling form.

```
Access:Inventory.UseFile(UseType>Returns)
```

Indeed, this is so. See the demo app, [Multiple Forms | Second Form \(Corrected\)](#). The addition of this single line of code in the procedure's `INIT` method prevents the `FileManager` from trashing the buffer changes when the second form closes.

Now the use of a widow to supplement a form does what you expect (and want) it to do.

## Summary

It is nice to know that I have the typical (for Clarion) three choices in how to do something (spread data entry across multiple pages, in the current case). I particularly like the EZ wizard, created at runtime, and, I think, many cases in which a window is used can be replaced with runtime wizards.

Rather more important is to understand how data moves from disk to memory (the buffer) and, thence, from window to window. In fact, data moves down this chain just as one would expect. *By design*, it does *not* move back up the chain unless you explicitly tell the `FileManager` to do so. Sometimes the templates just don't read your mind very well at all.

[Download the source](#)

---

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

## Reader Comments

[Add a comment](#)

**Interesting article. I like the concept of using a wizard...**

**Thanks for pointing out the problem with the Record Buffer...**

**Now I know why a customer's app trashes the record buffers!...**

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](#)

[Advanced Search](#)

[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)

**Related Categories**

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

[Topics](#)

**Related articles**

## Creating A SCADA Interface With Clarion (Part 2)

[Advertise](#)

Published 2002-03-21   [Printer-friendly version](#)

[Reader](#)

[Comments](#)

[Last week](#) I described how one of the companies I work for uses a Clarion application for Supervisory Control and Data Acquisition (SCADA) industrial automation system. I looked at some of the data structures and protocols. Now it's time to look at the system that displays status information to the user.

[Call for Articles](#)

[Log In](#)

[Membership/ Subscriptions](#)

When the user clicks a Current Status button, the window shown below opens.

[FAQ](#)

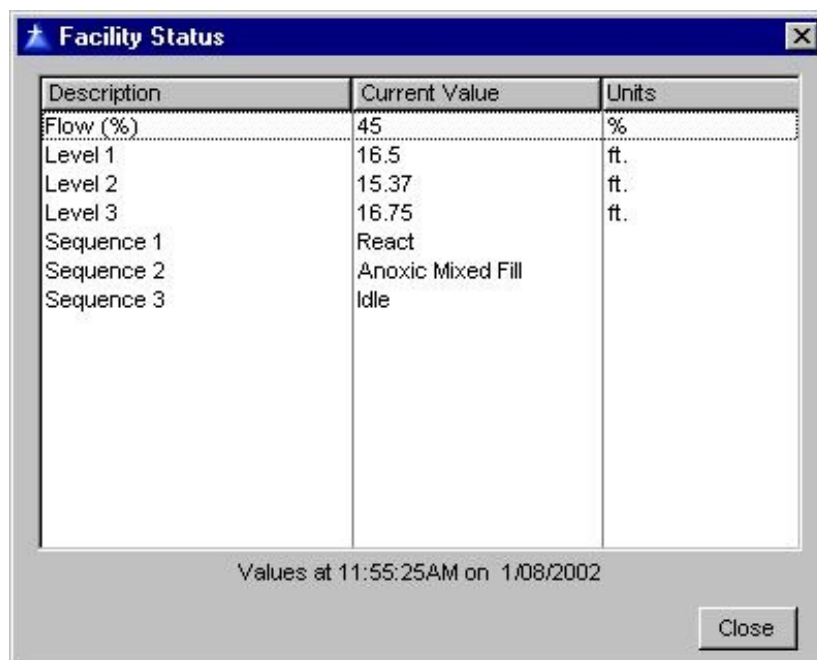
[Privacy Policy](#)

[Contact Us](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)



**Figure 4. Status Display Window**

This window just contains an unformatted list populated without the control template and a Close button. The `TIMER` attribute of the window is set to 1500 so the values will be



refreshed every 15 seconds. The FROM attribute of the list is set to ElementListQ, which is defined below:

```
ElementListQ    QUEUE,PRE(ELQ)
Description     STRING(100) ! Same as ELE:Description
CurrentValue    STRING(50)  ! Current value from PLC
Units          STRING(20)  ! Same as ELE:Units
ElementID      LONG        ! Same as ELE:ElementID
END
```

After the files are opened, the following code fills

ElementListQ:

```
ElementCount = RECORDS(ChosenQ)
LOOP WhichElement = 1 TO ElementCount
    GET(ChosenQ, WhichElement)
    ElementListQ ::= ChosenQ
    ADD(ElementListQ)
END
Access:plclog.UseFile()
DO UpdateQ
```

The UpdateQ routine fills in the data values from the newest record in the plclog file and updates the text at the bottom of the screen:

```
UpdateQ ROUTINE
    SET(plclog)
    IF Access:plclog.Previous() = LEVEL:Benign THEN
        ElementCount = RECORDS(ElementListQ)
        LOOP WhichElement = 1 TO ElementCount
            GET(ElementListQ, WhichElement)
            ChosenQ.ElementID = ElementListQ.ElementID
            GET(ChosenQ, ChosenQ.ElementID)
            ElementListQ.CurrentValue = NewGetData(OverMemo,TRUE)
            PUT(ElementListQ)
        END
        StatusText = 'Values at ' & FORMAT(PLC:LogTime, @T6) |
                    & ' on ' & FORMAT(PLC:LogDate, @D2)
    END
```

Note the line `ElementListQ.CurrentValue = NewGetData(OverMemo, TRUE)`. `OverMemo` is a Global `STRING(255)` declared `OVER(PLC:PLCData)`, the memo field in the `plclog` file that contains the data block. Redeclaring it as a string allows the programmer to use string slicing to access the data. The data conversion process also uses this technique. The `NewGetData` function listed below (Listing 5) does the data conversion. I've added line numbers for easier reference in the discussion following the listing.

### Listing 1. NewGetData Function

```

! Prototype
NewGetData FUNCTION(*STRING, SHORT),STRING
! Definition
NewGetData PROCEDURE (RawData, ForStatus) ! Declare Procedure
FirstByte      USHORT      ! convert word offset to byte offset
HoldReal       SREAL       ! hold SREAL data
RealAsString   STRING(4),OVER(HoldReal) ! allow string slicing
HoldShort      USHORT      ! hold binary data
ShortAsString  STRING(2),OVER(HoldShort) ! allow string slicing
HoldDec        DECIMAL(4)  ! hold BCD data
DecAsString    STRING(3),OVER(HoldDec)  ! allow string slicing
ResultString   STRING(50)  ! return value
BarAt          LONG        ! separator location in ELE:StatusText
CODE
1  IF ChosenQ.Word <= PLC:BytesUsed/2 THEN
2      FirstByte = ChosenQ.Word*2-1
3      IF UPPER(ChosenQ.RealNumber) = 'Y' THEN
4          RealAsString = RawData[FirstByte+3] |
                    & RawData[FirstByte+2] |
                    & RawData[FirstByte+1] |
                    & RawData[FirstByte]
5          RETURN(ROUND(HoldReal, ChosenQ.RoundTo))
6      ELSE
7          IF ChosenQ.Bit > 0 THEN
8              ShortAsString = RawData[FirstByte+1] |
                        & RawData[FirstByte]
9              HoldShort = BAND(HoldShort, |
                        BSHIFT(1,ChosenQ.Bit-1))
10             IF HoldShort > 0 THEN
11                 HoldShort = 1
12             END
13             IF (NOT ForStatus) |
                AND UPPER(ChosenQ.ReversedOnGraph) = 'Y' THEN
14                 HoldShort = BXOR(HoldShort, 1)
15             END
16             IF ChosenQ.StatusText <> '' AND ForStatus THEN
17                 BarAt = INSTRING('|', ChosenQ.StatusText, |
                            1, 1)
18             ELSE
19                 BarAt = 0
20             END
21             IF HoldShort > 0 THEN
22                 IF BarAt > 0 THEN
23                     ResultString = |
                                ChosenQ.StatusText[BarAt+1 |
                                : LEN(CLIP(ChosenQ.StatusText))]
24                 ELSE
25                     ResultString = '1'
26                 END
27             ELSE
28                 IF BarAt > 1 THEN
29                     ResultString |
                                = ChosenQ.StatusText[1 : BarAt-1]
30                 ELSE
31                     ResultString = '0'
32                 END
33             END
34         ELSE
35             DecAsString = CHR(0) |
                        & RawData[FirstByte : FirstByte+1]
36             ResultString = ROUND((HoldDec |
                                * ChosenQ.Multiplier), |

```

```

                                ChosenQ.RoundTo)
37         IF ChosenQ.GroupID > 0 AND ForStatus THEN
38             MES:GroupID = ChosenQ.GroupID
39             MES:MessageID = ResultString
40             IF Access:messages.Fetch(MES:byGroupMessage) |
                = LEVEL:Benign THEN
41                 ResultString = MES:Message
42             END
43         END
44     END
45     RETURN(ResultString)
46 END
47 ELSE
48     ResultString = '0'
49     RETURN(ResultString)
50 END

```

The second parameter passed to `NewGetData` is set to `FALSE` if the results are to be graphed instead of listed (I may write another article describing the graphing part of the application). Line 1 checks to make sure the referenced word actually exists in the data block. Line 2 converts the word offset to a byte offset. Line 3 checks to see if the element is a real number. If it is, Line 4 fills in the `SREAL`'s four bytes using string slicing on the `OVERED` variable. Notice that *the order of the bytes has to be reversed!* No, I don't remember why Intel chose to store numbers this way ("Big Endian" as opposed to "LittleEndian," if I remember correctly). Line 5 rounds the number as desired and returns the value.

If the element is not a real number, line 7 checks to see if it is a binary element. If it is, line 8 copies the word's bytes to a `SHORT` using string slicing. Again, the bytes are reversed. Line 9 masks off the other bits in the word. If the result is greater than 0 that means the bit was set, so lines 10-12 set the value to 1. Lines 13-15 take care of reversing the logic if the result is going to be graphed and the `ReversedOnGraph` flag is set. Lines 16-33 choose the appropriate text from `StatusText` if the result is not going to be graphed.

If the element is not real or binary, it must be BCD, so line 35 fills in the three bytes for the `DECIMAL`. Since Clarion's `DECIMALS` have a sign nibble, three bytes are required for the `DECIMAL(4)` instead of just two. The first byte is set to zero to indicate a positive number and an unused digit. The second and third bytes are filled in from the data block. Notice that the order of the bytes here is *not* reversed! (Isn't this fun?) Line 36 sets the decimal point and rounds as desired. If the `GroupID` is set, lines 37 through 43 fetch the appropriate text from the



messages file.

Line 45 returns the result for the binary and BCD cases. Lines 48-50 return a zero if the word offset specified was larger than BytesUsed.

After the window opens, a property assignment set the format of the list:

```
?ElementList{PROP:Format} = |
    & '100L(1)~Description~L(1)@s100@|M' |
    & '80L(1)~Current Value~L(1)@s20@|M' |
    & '24L(1)~Units~L(1)@s20@'
```

Finally, in the Timer event embed for the window, after the generated code, the values are updated by calling the UpdateQ routine.

```
StatusText = 'Updating...'
DISPLAY(?StatusText)
DO UpdateQ
DISPLAY()
```

## Implementation

This logger and viewer application pair started operating over five months ago. During that time several new features were added to each, including the graphing I mentioned earlier. Also, a touch screen ([EZTouch](#), connected directly to the PLC) mounted on the control panel in the plant allows the operators to observe the system's status without going back to the office. So far, it works pretty well. Occasionally unusual circumstances such as high flow or equipment failure confuse the part of the PLC's program that deciphers what sequence the current system has selected. That doesn't happen very often, however, and the operators can easily follow what the system is doing by looking at the PC in the office or the equipment status on the control panel.

## Summary

This project demonstrates that Clarion is a great tool for SCADA interface applications. Some features that are especially useful are the OVER declaration, the EVALUATE function, string slicing, and bitwise functions. These low-level language components combined with the high-level database access components give Clarion great flexibility. Third party

tools such as WinEvent make adding both low-level and "neato" features easy. Clarion's template language is another benefit; the dual list box interface for choosing multiple items from a list is a good candidate for a control template.

## Reader Comments

[Add a comment](#)

**[Links to products mentioned: www.capesoft.com >...](http://www.capesoft.com)**

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.

[Home](#)   [COL Archives](#)

[Advanced Search](#)

[Browses](#) · [COM/OLE](#) · [Databases](#) · [Design & Development](#) · [Forms](#) · [Getting Started](#) · [Internet](#) · [Misc.](#) · [Multimedia](#) · [News](#) · [Non-tech](#) · [OOP](#) · [Reports/Processes](#) · [Reviews](#) · [Templates](#) · [Tips/Techniques](#) · [Using ABC](#)

**Related Categories**

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

[Topics](#)

**Related articles**

## Baby Overrides Publishing Schedule!

[Advertise](#)

Published 2002-03-22   [Printer-friendly version](#)

[Reader](#)

[Comments](#)

The next issue of Clarion Magazine will appear the week of April 8, 2002. Clarion Magazine's publisher, Dave Harms, is taking some time off to be with his wife Bonny and their newborn daughter Sophia Elizabeth Nicole. Sophie came into the world at 9:03 p.m. Central Time on the first day of spring, March 20, 2002. She weighed in at 7 pounds 4 ounces, and is, of course, as cute as a button. Both mother and daughter are doing great!

[Call for Articles](#)

[Log In](#)

[Membership/ Subscriptions](#)

[FAQ](#)

[Privacy](#)

[Policy](#)

[Contact Us](#)

### Reader Comments

[PDFs](#)

[Add a comment](#)

[Freebies](#)

[Open Source](#)

- [\*\*Congratulations Dave! Those are great news! You just...\*\*](#)
- [\*\*Congradulations Dave on the birth of your new baby. It's...\*\*](#)
- [\*\*Congratulations, Dave and Bonny! Isn't it the most...\*\*](#)
- [\*\*Congratulations David Heres to Many restless nights...\*\*](#)
- [\*\*Congratulations! Little girls are the best! Mine is...\*\*](#)
- [\*\*Congratulations! Such wonderful news... and I certainly...\*\*](#)
- [\*\*Congratulations Dave! All the best for ALL of...\*\*](#)
- [\*\*Great news Dave! Welcome to the club . Best wishes from...\*\*](#)
- [\*\*Great news Dave! Welcome to the club . Best\*\*](#)



**whishes from...**

**Well done to all of you! How special that must feel for...**

**Pictures? Just one or two? I'm glad to hear that mother...**

**Congrats Dave and Bonny, hope all are doing well.**

**Way...**

**Thanks, everyone! Baby Sophie is \*very\* impressed by all...**

**Congratulations Dave and Bonny! I know how thrilling this...**

**Welcome, Sophia!**

**Congratulation Dave for being a father.....! Hope Sophie...**

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.