Clarion Magazine

Home COL Archives

Accessing TPS Files Via ASP

In this article Brian Staff provides an introduction to Active Server Pages, or ASP, for Clarion developers. With ASP you can use the TopSpeed ODBC driver to display a TPS file (or any ODBC compliant data) in a browser.

Posted Wednesday, May 08, 2002

Look Out, It's Outlook! (Part 1)

The subject of interfacing with MS Outlook comes up regularly on the newsgroup. Mark Riffey takes a second look at his Outlook code, and discovers unexpected help in MS Word. Part 1 of 2.

Posted Thursday, May 09, 2002

Multi-DLL Debugging/Profiling

The cciProfiler and cciDebug classes are a great aid to debugging and understanding what ABC is doing under the hood, but until now were limited to single EXEs. Here are Matías Flores' modifications to allow these classes to be used in multi-DLL applications.

Posted Friday, May 10, 2002

Look Out, It's Outlook! (Part 2)

The subject of interfacing with MS Outlook comes up regularly on the newsgroup. Mark Riffey takes a second look at his Outlook code, and discovers unexpected help in MS Word. Part 2 of 2.

Posted Tuesday, May 14, 2002

Data Structures and Algorithms Part II - The List

How many times have you been asked to write a list? Lists are everywhere: there are shopping lists, to do lists, lists of people's contact details, and lists of appointments. The List is so pervasive and so valuable that many people have spent many years discovering how to effectively represent this structure in computer memory with the greatest efficiency.

Posted Friday, May 17, 2002

ETC III Is Under Way!

ETC III is now underway! Here are some early pics from registration and the pre-conference sessions. Check back regularly with Clarion Magazine

News

USB Print/Fax/Email From DOS Apps

DOS Printer

EasyWinIco 1.00 Freeware

0-MeZz Bugfix

gFileFind Contest Ends Friday, May 31, 2002

Sterling Prize Drawing Deadline May 31, 2002

Office Templates v2.0 Beta

VideoCapture Class Wrapper Released

Das ClarioNET Upgrades

INN Bio for 23-May-2002

Icetips Professional Wizards Released

Pre-Release Clarion 5.508(h) And CLarioNET 1.2 Available

0-MeZz Templates Released

Softmasters ETC Discount

XPMenu Lookalike Template Added

Office Templates v2.0 Available Soon

RichReport Beta 3

0-HaZzle Gold

PDF-Tools Evaluation Version

SealSoft's ETC-III Special Discount

Product Scope 32 PRO Version 4.5 Released

EasyAnimation 1.0 Released

Application Shell May Special

Purchase XPMenu Online

for more pictures and in-depth conference reporting.

Posted Tuesday, May 21, 2002

ETC III: Keynote Address - Bob Zaunere

Bob Zaunere has given the keynote address at ETC III. Clarion Magazine's Tom Hebenstreit was there and files this report.

Posted Wednesday, May 22, 2002

ETC III: CapeSoft NetTalk - Bruce Johnson

ETC this year is featuring a group of extra sessions, hosted by vendors of various third party products. Kicking it all off was a two-hour talk given by Bruce Johnson of CapeSoft that focused on their NetTalk TCP/IP communications accessory.

Posted Thursday, May 23, 2002

ETC III: More Pics, Including The Cajun Feast

More pictures from ETC III, including the legendary Cajun feast, this year bigger and better than ever.

Posted Thursday, May 23, 2002

ETC III: OOP - It's Not POO Spelled Backwards

Russ Eggen had the difficult task of filling the gap between Wednesday's lunch and the much-anticipated Cajun cookout. His plan - to fill the afternoon with the basics of OOP so that any non-OOPer would be at least cognizant of classes, methods, properties and interfaces before heading out to the mountains for some etouffee. Mark Riffey reports.

Posted Friday, May 24, 2002

There Will Be An ETC IV

ETC organizer Lee White has confirmed that yes, there will be an ETC IV, probably in early June of 2004.

Posted Saturday, May 25, 2002

ETC III: A Cajun's View

ETC's resident Cajun cook and Clarionite Andrew Guidroz offers a personal perspective on ETC III.

Posted Monday, May 27, 2002

ETC III: Photos By Mark Riffey

Browse these ETC-III photos by Mark Riffey - you can also add your own comments to individual photos.

Posted Tuesday, May 28, 2002

ETC III: Photos By Andrew Guidroz

Photos from ETC by Andrew Guidroz, including the reception, sessions,

Nice Touch Solutions ETC-III Special

Linder Software Office Closed May 23-26, 2002

Gitano ETC 20% Off Special

CapeSoft ETC Madness Specials

Purchases of BigTamer(tm) During ETC-III

Outlook Import Template Updated

Coollook Template 1.1 Available

Free Software From C&G

INN Bio For 16-May-2002

JT Free Progress Bar Replacement Template Updated

C5.6 Info And Clarion ASP Example Applications

IceTips Software Office Closed May 28 To June 12, 2002

Solace Software Holiday

SealSoft's xWhatsNew v1.0

RichReport Beta 2 Released

ClarioNET Deployment Manager

Native Clarion XPMenu Lookalike

RInstall Version 1.c (Beta) Available

Legacy To ABC Template Bundle Price

XPMenu Product

File Syntax Change Template

INN Bio for 8-May-2002

Icetips Pro Wizards

Batch Email And Save Report

SoftMasters Controls Pack 1.0 Released

EasyMultiTag 2.01 Released

Windows XP Compatibility Update

RichReport Beta 1.0

VizAcc Screen Capture Beta

The Wizzie Group - Call For Second Hand Wizatron Resources

Cajun cookout preparations (daytime pictures of the awesome cookout location), and heading home. Add your comments!

Posted Tuesday, May 28, 2002

ETC III: Photos by Brad Noble (1)

Scenes from the conference sessions, attendees in more relaxed surroundings, and more Cajun cookout pics. Add your comments!

Posted Tuesday, May 28, 2002

ETC III: Photos By Brad Noble (2)

Lots of photos of people eating and relaxing, plus a few of the sessions. Add your comments!

Posted Tuesday, May 28, 2002

Thanks! From Lee White & KCUG

A note of thanks from Lee White and the Knoxville Clarion Users Group.

Posted Tuesday, May 28, 2002

ETC III: ClarioNET and More - Bob Foreman

When Bob Zaunere finished his ETC III keynote address, two hours longer than scheduled and light years beyond Nik Johnson's expectations with respect to content, there seemed to be little left for Bob Foreman to add. But while the ASP templates have been announced before, and ClarioNET has been available for some time, Bob's presentation added substance that only an actual demonstration can provide.

Posted Thursday, May 30, 2002

ETC III: Programming in Clarion's ABC - Bruce Johnson

Bruce has expanded his previous presentation at ETC 2000 to cover significantly more information. For example, attendees of this session received a 137 page book, written by Bruce Johnson, that details all the information discussed. The book for this year's session is approximately three times bigger than before. Brice Schagane reports.

Posted Thursday, May 30, 2002

ETC III: Reports: Paying the Price of Flexibility - Steve Parker

Steve Parker presented the sixth and final session of ETC-III conference. The author of more than 100 articles on Clarion, Steve has a lot of experience with reporting. Brice Schagane provides a summary.

Posted Thursday, May 30, 2002

Looking for more? Check out the **site index**, or **search the back issues**. This site now contains more than 700 articles and a total of over a million

Duke Application Shell May Special

New SysViewPack Bundle

twTools 1.7 Released

INN Bio for 2-May-2002

NetTalk Goes Gold

NetTalk At ETC

Free PowerSearch License

KSprbar 1.20 Released

Bug Poster Source Updated

Search the news archive

words of Clarion-related information.

Clarion Magazine

Home COL Archives

Topics > Internet > ASP

Accessing TPS Files Via ASP

by Brian Staff

Published 2002-05-08

I'm guessing that most of you have spent years developing Clarion applications for your clients, where their data is stored in TopSpeed (TPS) files. I'll also wager that you've been asked more than once to explore the possibility of displaying some of that data on an internet/intranet web page.

This article is designed to introduce you to a new development environment: Active Server Pages, or ASP. With ASP you can use the TopSpeed ODBC driver to display a TPS file (or any ODBC compliant data) in a browser, for your client.

ASP files typically contain VBScript, which is really easy to read if you can read Clarion and HTML. The VBScript I'll describe will use ADO methods/properties and an SQL statement to read the data from the TPS file(s). You could also use JavaScript or PerlScript, but VBScript is easier to learn, although it's not as rich as JavaScript.

Firstly a word about the contents of ASP files. You do not need any special third party software to generate ASP pages - they are pure text. All you need is an editor. I use my C55 editor (I like CTRL-2 and CTRL-Y). These ASP files are not that different in their function from Clarion's own template (TPL/TPW) files. TPL/TPW files have a mix of template language statements and Clarion language statements. In ASP files, VBScript and HTML are intermixed too. In TP? files, you can identify template language by the # character on the front of each statement. In ASP files, server side VBScript code (that is, code that executes on the server before the page is sent to the client) is inside a `<%...%>` structure. The code outside of those structures is pure HTML tags, and is simply passed through to the client.

To test ASP on your own machine you'll need to install IIS (Internet Information Services) or PWS (Personal Web Service, included with MS FrontPage). You will need to set up a root virtual directory, which by default is \inetpub\wwwroot.

Your machine can now be both the client and the server. Launch ASP files in your browser by typing in a URL in this format:

```
localhost/listcust.asp
```

The localhost identifier is a special server name identifying your own (local) machine, and it resolves to an IP address of 127.0.0.1. Do not use a syntax of c:/yourDirectory/listcust.asp - that will merely try to copy the contents of the ASP page to the client and it will not invoke the IIS process. You'll get a similar symptom if you double-click an ASP page in Explorer.

When you enter the correct URL, your browser sends that URL from the client to the server using the HTTP protocol. IIS (or PWS) on the server receives that request and loads the requested ASP pages and any INCLUDEd files (I'll talk about these a little later). Once the file(s) have been loaded on the server, IIS then executes the server-side script to generate an HTML page, which is usually a combination of the HTML in the ASP page and any HTML which is generated by the server-side VBScript code. This page is then sent back to the client to display in the browser. Server-side scripts are never sent to the client, however client side scripts are sent to the client. More on those later too.

It is beyond the scope of this article to explain every piece of VBScript, however I do want to identify one line that will allow you to access your TopSpeed data: the connection string assignment. I will not be showing how to use data sources as defined in the ODBC administrator, but rather a direct connection string.

All ODBC drivers have their very own connection parameters. The TopSpeed version in its simplest form is like this:

```
strConnect = "DRIVER={Topspeed ODBC Driver};" |
 & "DBQ=c:\yourDirectory\;PWD=;"
```

I was unable to use the "developer" version of the TopSpeed ODBC driver in an ASP page. I suspect the cause is the driver splash window. Maybe SoftVelocity could fix that problem for all of the developers wishing to experiment with ASP.

Carl Prothman, a Microsoft Visual Basic MVP has compiled a list of ODBC connection strings for a number of applications and databases, including Access, Excel, MySQL, SQL Server, Sybase, and many more.

Here's an example of real-world TopSpeed ODBC driver connection string:

```
strConnect = "DRIVER={Topspeed ODBC Driver};DBQ=" |
  & Server.MapPath("./")    & "\;PWD=;"
```

ASP will convert the root virtual directory to a real directory string using `Server.MapPath` and then add a "\" on the end. The first step is to make a connection to the TPS directory

```
Set objConn = Server.CreateObject ("ADODB.Connection")
objconn.CursorLocation = adUseClient
objconn.Mode = adModeRead
objConn.Open (strConnect)
```

The connection string using the `DBQ` value points to the data directory containing your TPS files, and the `PWD` value identifies a null password. You do not have to point to a specific TPS file with a TPS connection string. You merely point to a directory which contains one or more TPS files. The specific file you wish to read is identified in your SQL statements, i.e.:

```
sql = "SELECT * FROM customer"
```

In the above statement `customer.tps` is the TopSpeed data file. You could also use the following syntax:

```
sql = "SELECT * FROM ""customer.tps"" "
```

Next, use an ADO connection method to get a recordset, which is all the rows and columns of data specified by the SELECT statement:

```
set oCust = objConn.execute(sql)
```

Once you've retrieved the recordset from the TPS file, it's fairly simple to iterate through the individual records using the following syntax:

```
While Not oCust.EOF
  ...
  oCust.MoveNext
Wend
```

Then it's a matter of writing the HTML tags so that the data is formatted appropriately for the web page, which is then sent to the client. Be sure to use the View|Source option in IE or Netscape to see exactly what was sent to the client browser.

ASP objects

ASP, which is an environment, not a language, offers just six built-in objects, which the server-side scripts can use:

Object	**Description**
`Request`	Gets information from the user (client) that's passed in a HTTP request.
`Response`	Used to send information back to the client.
`Server`	Controls server-side activity, like creating component objects.
`Session`	Stores information about the current user session.
`Application`	Stores information for the entire lifetime of the application.
`ObjectContext`	Used to access the Microsoft Transaction Server system.

The most frequently used statement (response.write) is used to send text directly to the generated HTML file.

Included in the downloadable source for this article is the LISTCUST.ASP file, which you can see in action on the web by submitting the following URL in your browser:

```
http://www.scoreboard.to/listcust.asp
```

The file contains about 70 lines of code in this file, which is fairly easy to read. It uses the samples above to output a simple HTML table from a TopSpeed file.

It is important to note that whenever a VBScript variable is primed with an object, usually using the SET statement as in the following syntax:

```
set oCust = objConn.execute(sql)
```

In the above case that variable must be cleared after you have finished with it. This is to avoid memory leaks, and it is done with the following statement:

```
set oCust = nothing
```

LISTCUST.ASP

Here are the basic building blocks for inside the LISTCUST.ASP file. The actual file does contain some more padding like comments and error checking. This first part describes the language to be used for the server-side scripts, and I will not allow implicitly declared variables.

```
<%@ language="VBScript" %>
<%
```

```
  Option Explicit
%>
```

Here is where I include any other files I will be needing, the first one being an equates-type file for ADO.

```
<!-- #INCLUDE FILE="adovbs.inc" //-->
<!-- #INCLUDE FILE="stt.inc" //-->
```

Here I declare the beginning of the HTML file to be sent back to the client.

```
<html>
<head>
</head>
<body style="text-align: center; background-color: silver;">
```

Now, I revert to server-side scripting and declare some VBScript variables, define the connection string and open an ADO connection object:

```
<%a
 dim objConn
 dim oCust
 dim sql
 dim strConnect
 strConnect = "DRIVER={Topspeed ODBC Driver};DBQ="
   & Server.MapPath("./") & "\;PWD=;"
 Set objConn = Server.CreateObject ("ADODB.Connection")
 objconn.CursorLocation = adUseClient
 objconn.Mode = adModeRead
 objConn.Open (strConnect)
```

Here is the sql statement to be used inside the execute method of the connection object which will return a recordset object.

```
sql =  "SELECT "         &_
       " sysid,"         &_
       " company,"       &_
       " city,"          &_
       " country "       &_
       "FROM "           &_
       " customer "      &_
       "WHERE "          &_
       " sysid = sysid " &_
       "ORDER BY "       &_
       " country, "      &_
       " city, "         &_
       " company "       &_
       " "
 set oCust = objConn.execute(sql)
```

Next, I define the HTML table and specify the table headers.

```
response.Write "<table border='2' cellspacing='0'
  cellpadding='3' width='auto'
  style='background-color: #eeeeee;'>" & vbCRLF
response.Write "<caption>List of Customer Records</caption>"
  & vbCRLF
response.Write "<tr style='background-color: gray;
  color: white;'>" & vbCRLF
response.Write "<th>Company</th><th>City</th><th>Country</th>"
  & vbCRLF
response.Write "</tr>" & vbCRLF
```

I now iterate through the recordset rows, and create the HTML for each row in the HTML table:

```
While Not oCust.EOF
  response.Write "<tr>"
  response.Write "<td>" & oCust("company").Value
    & "&nbsp;</td>"
  response.Write "<td>" & oCust("city").Value
    & "&nbsp;</td>"
  response.Write "<td>" & oCust("country").Value
    & "&nbsp;</td>"
  response.Write "</tr>" & vbCRLF
  oCust.MoveNext
Wend
```

At the end of the recordset, the HTML table must be closed, and the objects cleaned up:

```
 response.Write "</table>" & vbCRLF
   oCust.Close
   objConn.Close
   clearDims()

 sub clearDims()
   Set oCust = Nothing
   Set objConn = Nothing
   end sub
%>
```

Lastly, I must declare the end tags for the HTML file:

```
</body>
</html>
```

That's it! It's really quite simple. The `vbCRLF` is added to some of the generated code so that the resultant HTML code that gets sent to the client is nicely formatted.

The actual LISTCUST.ASP file contains just enough clues to make anyone familiar with Clarion dangerous on the internet. Be my guest, and modify it for your own purposes. The most common coding mistakes you will likely make are using the wrong quotes character to identify a string, and omitting the word `THEN` at the end of an `IF` statement line.

In the downloadable source at the end of this article I have included a GLOBAL.ASA file which contains some optional event handlers for the application, and ADOVBS.INC, which is an EQUATES-type file for ADO. Both of these need to be in the virtual root directory. I have also included an empty STT.INC file for now. I'll cover the real contents in the next article, which will also deal with threading and some further methods/properties in ADO access your data. Ordinarily, multiple client requests to the same ASP page at the same time is not a problem, but the TopSpeed driver seems to have a weakness with windows threading. STT.INC will make it possible to throttle the requests to a single user at any one time.

If you happen to get the dreaded ASP 0115 (table not found) error displayed in your browser window, you will have to restart IIS or ,if you're using PWS, reboot your machine...sorry about that! It appears to be a TS ODBC driver problem. This is almost certainly caused by your connection string not pointing to the correct directory containing your TPS file.

I encourage you to learn HTML thoroughly. The specifications can be found at http://www.w3.org/TR/html401/ and there is an on-line validator available at http://validator.w3.org/ which will check for well-formed HTML. You can either submit your URL or upload an HTML file that site.

Finally, here are some further resources for ASP:

- http://www.w3schools.com
- http://www.aspfaq.com/
- http://www.infinitemonkeys.ws/infinitemonkeys/
- http://www.15seconds.com

Download the source

Brian Staff was born and raised in Rugby, England, and lived for 28 years in Vancouver, Canada. He worked too many years for Honeywell on mainframes, and spent six years as an independent developer, including four years developing software for DirecTv. Brian currently develops point of sale systems and web applications for JDA Software in Phoenix, Arizona (it's a dry heat). A member of Team Topspeed since Feb 1996, Brian is also the author of the Xplore templates and is a coach and volunteer web site developer for the local soccer community. He is married to Valerie, has three soccer-playing daughters, and is a former international level rugby referee.

Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > COM/OLE > COM/OLE

Look Out, It's Outlook! (Part 1)

by Mark Riffey

Published 2002-05-09

A frequent question on the newsgroups is one of "how do I get <whatever> from Outlook", or vice versa. I've been involved in newsgroup threads on this topic a time or two and eventually ended up digging around in MSDN because a definitive answer never really came out of those threads. Okay, the definitive answer may have come out of those threads, but the code itself didn't appear in the thread, so this confirmed member of the LPS (Lazy Programmer's Society) got sidetracked and ended up doing something else for a while.

Recently, "dealing with Outlook" floated back to the top of my ToDo list. About that same time, someone posted some code on the newsgroup that he used to get contacts from Outlook. It turned out that it was frighteningly similar to the code I had ended up with when I last stopped fiddling with Outlook, which piqued my interest in what little I had managed to get working.

There is a good reason that our code was similar. Talking to Outlook via COM requires a specific sequence of events, and there is little you can do to escape that. So, let's not fight Bill G, let's just put up with him for a bit.

A treasure MAPI

The key to the treasure is knowing where in MSDN to find the way in the front door. In this case the front door is the MAPI NameSpace. The common name for the MAPI NameSpace is "Outlook's database". Accessing the name space and its contents is discussed in minute detail in the MSDN articles that explain the Outlook object model. Use those articles, but don't get attached to them. There is a much faster way.

Once you get this far, dealing with Outlook is quite simple. In English, it goes like this:

- Create a COM object.
- Open the NameSpace.
- Open the Contacts folder
- Add/Delete/Find Contacts.
- Release the resources associated with the items you opened or created above, in the reverse order that you created them.

Listing 1 shows some sample code that you might use when importing contacts from Outlook via a Clarion process.

Listing 1. Importing contacts from Outlook

```
!To turn on Clarion's OLE error messages
! use = 1 below.
?OLE{prop:reportexception} = 0
MyOLE = ?OLE{prop:create}
myNameSpace = ?OLE{'GetNameSpace("MAPI")'}
loc:status = 'Opening Outlook name space'
display
loc:status = 'Opening Outlook contacts folder'
display
myContactFolder = ?OLE{clip(myNameSpace) |
  & '.GetDefaultFolder(10)'}

! see the notes re: the MS Word object browser
!  to find the folder numbers you need
loc:status = 'Contact folder opened'
display
j# = ?OLE{clip(myContactFolder) & '.Items.Count'}
loc:status = 'There are ' & j# |
  & ' contacts in your Outlook'
display
free(q:ol)
?progress1{prop:rangelow} = 1
?progress1{prop:rangehigh} = j#
?progress1{prop:progress} = 0
stream(prospect)

LOOP i# = 1 to J#
   clear(q:ol)
   myContact = ?OLE{clip(myContactFolder) |
     & '.Items(' & i# & ')'}
   if clip(?OLE{clip(myContact) & '.lastname'}) > ''
     Q:lastname    = clip(?OLE{clip(myContact) |
     & '.lastname'})
   elsif clip(?OLE{clip(myContact) & '.fullname'}) > ''
     Q:lastname    = clip(?OLE{clip(myContact) |
      & '.fullname'})
   end !if
   Q:firstname   = clip(?OLE{clip(myContact) |
      & '.Firstname'})
   q:company     = clip(?OLE{clip(myContact) |
      & '.Companyname'})
     Q:address1    = clip(?OLE{clip(myContact) |
        & '.businessaddressstreet'})
     !Q:address2    = clip(?OLE{clip(myContact) |
        & '.companyname'})
     Q:city        = clip(?OLE{clip(myContact) |
        & '.businessaddresscity'})
     Q:state       = clip(?OLE{clip(myContact) |
        & '.businessaddressstate'})
     Q:zip         = clip(?OLE{clip(myContact) |
        & '.businessaddresspostalcode'})
     Q:email       = clip(?OLE{clip(myContact) |
        & '.email1address'})
     Q:phoneH      = clip(deformat(?OLE{clip(myContact) |
        & '.hometelephonenumber'}))
     Q:phoneW      = clip(deformat(?OLE{clip(myContact) |
        & '.businesstelephonenumber'}))
     q:fax         = clip(deformat(?OLE{clip(mycontact) |
        & '.businessfaxnumber'}))
     q:dob         = deformat(clip(?OLE{clip(mycontact) |
        & '.birthday'}),@d17)
     if q:DOB> 200000 then q:DOB = 0.
    ! q:graduation  = clip(?OLE{clip(mycontact) & '.'})
     q:anniv       = deformat(clip(?OLE{clip(mycontact) |
        & '.anniversary'}),@d17)
     if q:anniv > 200000 then q:anniv = 0.
     !q:customerid       = clip(?OLE{clip(mycontact) |
        & '.customerid'})
     q:URL         = clip(?OLE{clip(mycontact) |
        & '.businesshomepage'})
     if clip(q:url) > ' '
     else
        q:URL      = clip(?OLE{clip(mycontact) |
        & '.personalhomepage'})
     end!if
     q:cell        = clip(?OLE{clip(mycontact) |
        & '.mobiletelephonenumber'})
     clear(pro:record)
     PRO:First_Name        =  Q:firstname
     PRO:Last_Name         =  Q:lastname
     pro:Company           =  q:company
     pro:ADDRESS           =  Q:address1
     pro:CITY              =  Q:city
     pro:STATE             =  Q:state
     pro:ZIP               =  Q:zip
     zip:zip_code          = pro:zip
     pro:Home_phone        =  Q:phoneH
     pro:bus_Phone         =  Q:phoneW
     pro:Faxphone          =  q:fax
     pro:Email             =  Q:email
     pro:DateOfBirth       =  q:DOB
     pro:anniversary       =  q:anniv
     pro:cellphone         = q:cell
     pro:url               = q:url
     access:prospect.insert()
    ?statusprompt{prop:text} = i# & ' of ' |
       & j# & ' contacts loaded so far'
     display
     ?progress1{prop:progress} = i#

   END
  ?progress1{prop:hide} = true
  ?statusprompt{prop:text} = ''
   display
   myContact{prop:release}
   myContact{prop:deactivate}
   mycontactfolder{prop:release}
   myContactFolder{prop:deactivate}
   myNameSpace{prop:release}
   MyNameSpace{prop:deactivate}
   ?ole{prop:release}
   ?ole{prop:deactivate}
   MyOLE{prop:release}
   MyOLE{prop:deactivate}
   setcursor()
   message ('Import is complete.')
```

Listing 1 contains fair amount of code, but most of it isn't exactly rocket science. It simply retrieves different fields from Outlook and moves them in to a record layout I am using. If you wish, you could do this quite differently, and most likely with more efficiency; the point is, you can do it easily.

Microsoft Word, your OLE assistant!

The good part about figuring out this stuff is that you don't have to wade through dozens of eye-reddening MSDN pages to find the details about where to go once you get the contacts folder open. Instead, you can use MS Word! Here's how I do it.

First, I open MS Word. Click Tools, Macro, Visual Basic Editor (or click Alt-F11) and the VB editor in Word is now open. I press F2 to open the object browser.

Sha-boom! (that's a technical term), I now have much of the Office and Windows object model in front of me. To see Outlook's model, I leave the Object Browser open and click Tools, References. I scroll down the list until I get to `Microsoft Outlook 10.0 Object Library`. Note that if I weren't working in Word XP, it might say `Outlook 9.0`. That's okay too. Next I check the box next to `Outlook xx.x Object Library`, then click OK.

Now I have the Outlook object model in the Object browser, so I click the drop down list that currently says "All Libraries", and select Outlook. The object browser now is showing everything I need in order to talk to Outlook. That's it. Wasn't that easy?

Okay, that's not it. But it's very close. Look at the code below:

```
myContactFolder = ?OLE{clip(myNameSpace) |
   & '.GetDefaultFolder(10)'}
```

You've read the article, now see the video!

Because the instructions to find all the stuff in Word's object browser might be a bit much for some without images, I made a [Camtasia video]() of my going there and finding stuff. The video frame size is 800x600. This is a two meg EXE (video and player, no install necessary).

Now look in the object browser for `NameSpace`and click on it. On the right, the browser shows all the properties, methods and objects in the `NameSpace` object. Find `GetDefaultFolder` and click it. The object browser shows you the syntax of the call to `GetDefaultFolder` down at the bottom.

Hold on a second…. What does that `(10)` mean? The answer is that `(10)` is an argument to `GetDefaultFolder`, and is a constant that identifies a specific folder in the MAPI name space. But how do I figure out what number to use? I use the object browser.

If I click on `olDefaultFolders` in that call, a list of the valid values for `olDefaultFolders` appears. One of those items is `olFolderContacts`. When I click on it, the green area at the bottom of the screen shows me the value is 10. Click. That was my head's light bulb turning on. The number 10 = contacts. If I click on another item, such as `olFolderCalendar`, I see that the value displayed is 9, so if I want to open the Calendar folder, I would use `(9)` in my `GetDefaultFolder` call. Woohoo, pass the boudin, I can see the light!

What's next? I need to figure out how to dig around in that folder. Since I got the folder name via a call to `GetDefaultFolder`, I need to figure out what methods and such I use to deal with folders. In a brief moment of clarity, I click on `Folders` in the left pane of the object browser, since `Contacts` is a member of the `Folders` collection. Sha-boom (again?). I see methods like `Add`, and `GetFirst`. I see properties like `Count`, which I use here:

```
j# = ?OLE{clip(myContactFolder) & '.Items.Count'}
```

In the import example in Listing 1 I retrieved the `Count` property of the `Items` object so I would know how many times to loop through the folder and grab contact names. How did I find out what the `Items` property contained? I clicked on it and pressed F1. Scary, eh?

Next, I needed to find out about the items in the contacts folder, so I looked for something related to contacts in the object browser. The guilty party? `ContactItem`. Just like before, when I clicked `ContactItem`, I got a nice list of properties and methods for the `ContactItem`. Now I'm ready to code. If I want the business street address for this contact, I write the following:

```
Q:address1 = clip(?OLE{clip(myContact) |
   & '.businessaddressstreet'})
```

So there I sit, coding based on information found in lowly MS Word, after numerous attempts at digging around in MSDN for the "secret" to dealing with Outlook. Do I feel stupid at this point? Sort of. But I got where I wanted to go, and that's the key.

Next time, I'll go over getting appointments and tasks, as well as finding items, adding them and updating them, but I'm guessing that you'll be well past that point by then.

Download the source

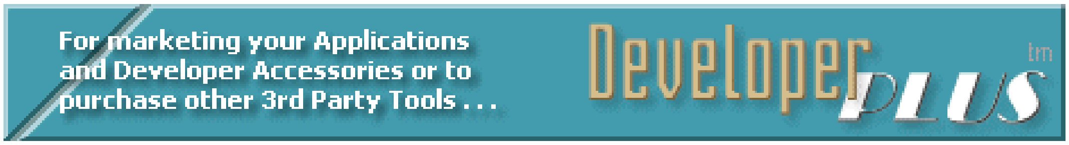
Mark Riffey has worked in the software industry, primarily in development and technical support for two internationally known enterprise software vendors, the world's premier information systems services company, a Fortune 100 manufacturer, and now Granite Bear Development. His business philosophy is simple: Be fair to your customers and yourself, surround yourself with brilliant people, work hard, be a good listener and have a little fun. Mark and his wife Jacki have two boys, Alex and Jonathan. Mark's other interests include Boy/Cub Scouting, backpacking/hiking and almost anything else outdoors, classic blues guitar, golf and photography.

Reader Comments

Add a comment

Mark, This is a great article. You saved my hours in...
Youre welcome, Hyrum. Thanks very much. Mark
Mark, A great article. Good to see an 'easy' guide to...
Thanks Devan, I appreciate the kind words. I just kinda...

Clarion Magazine


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

Home COL Archives

Topics > Tips/Techniques > Debugging

Multi-DLL Debugging/Profiling

by Matías Flores

Published 2002-05-10

When I read Dave Harms' profiling/debugging article for the first time I realized that I'd just found a very useful tracing tool. Until then I had no idea that there was a file called profile.clw on my libsrc directory. Unfortunately, I have to admit that I felt a little disappointed when I came to the last paragraph in the article:

> "Remember that profiling only works with single EXEs, not DLLs! That's a limitation of the runtime library, since the profiler hooks are not exported."

Ouch! That was not what I'd expected, since almost all my projects have several DLLs. So I decided to do something about it... but what?

The first thing I did was to test what really happens if I try to implement these classes in a multi-app project. So I took the dlltutor example application, added the global template to the four apps, and recompiled them. I was not surprised when I saw this example GPFing. I decided to see what could be done.

Version 1.3 improvements

In this article I'll only explain the changes that I've made to these classes. See the original ClarionMag articles for a detailed explanation of how the classes work. You can download the latest version (v1.3) of `cciDebugClass` and `cciProfilerClass` classes at the end of this page.

The first thing I needed to modify on the templates was to add the `dbg` object to the export list of the data DLL, and change the object declaration in the other apps to include the `EXTERNAL` and DLL attributes. Without these modifications, I would have four `dbg` objects (one local object for each application) instead only one global `dbg` object declared in the data application. For this purpose, I used the `%AddExpItem` group of the ABC template chain. This group receives a string and appends the `@?` characters to the end of that string. I called this group from the `%BeginningExports` embed point, an ABC embed point placed at the top of the .exp generated file.

```
#AT(%BeginningExports),WHERE(~%GlobalExternal)
#INSERT(%AddExpItem(ABC),'$'&%BaseDebugObjectName)
#ENDAT
```

```
#AT(%AfterFileDeclarations)
   #IF(%EnableProfiler)
     #IF(%GlobalExternal)
%[20 + (%CustomGlobalDataComponentIndent * 2)]BaseDebugObjectName↵
  %BaseDebugClass,EXTERNAL,DLL(_ABCDllMode_)
     #ELSE
%[20 + (%CustomGlobalDataComponentIndent * 2)]BaseDebugObjectName ↵
  %BaseDebugClass
     #ENDIF
   #ENDIF
#END
```

The next step was to add the `PROFILE:EnterProc` and `PROFILE:LeaveProc` procedures to the export list to make them available for being called from other applications. I added the `Init` and `Kill` methods of the `cciProfilerClass` for the same reason.

```
#AT(%BeginningExports),WHERE(~%GlobalExternal)
#INSERT(%AddExpItem(ABC),'$'&%BaseDebugObjectName)
#INSERT(%AddExpItem(ABC),'PROFILE:ENTERPROC')
#INSERT(%AddExpItem(ABC),'PROFILE:LEAVEPROC')
#INSERT(%AddExpItem(ABC),'KILL@F16CCIPROFILERCLASS')
#INSERT(%AddExpItem(ABC),'INIT@F16CCIPROFILERCLASS')
#ENDAT
```

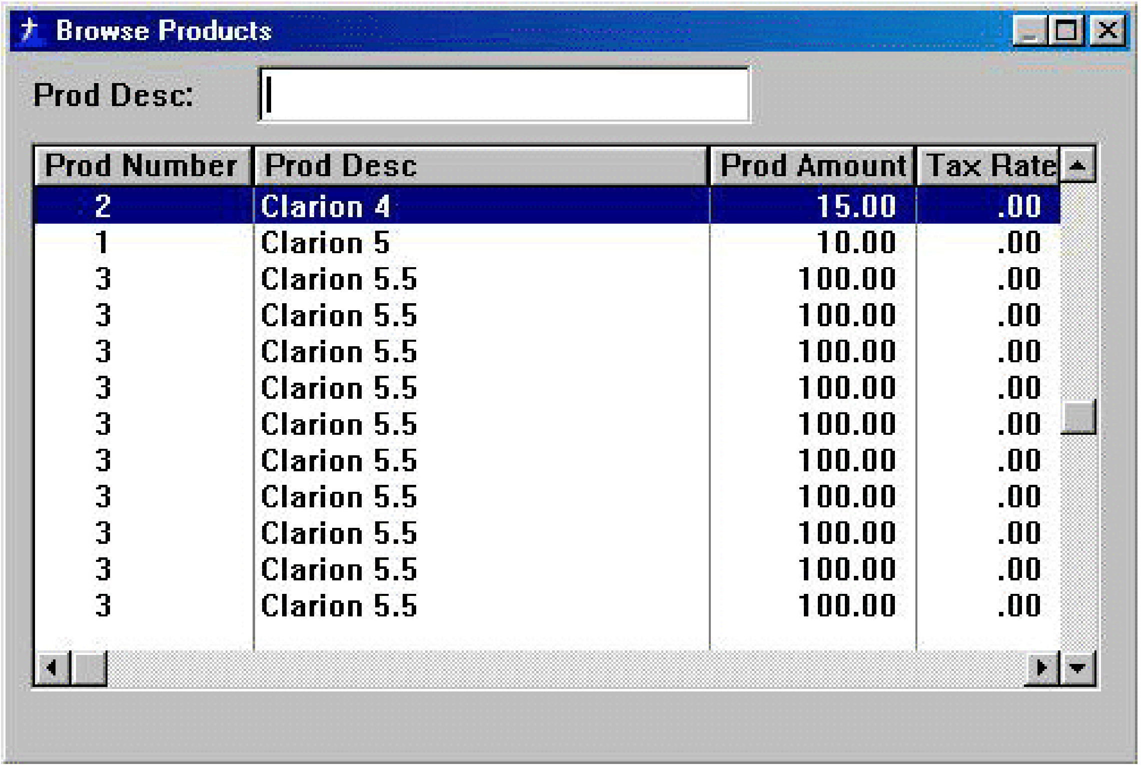
The last step was to add the cciprof.clw file to the project only if the app has the "Generate template globals and ABC's as EXTERNAL" option unchecked, to avoid the duplicate symbol error messages at the end of the linking process.

```
#IF(%BaseDebugClass = 'cciProfilerClass' AND ~%GlobalExternal)
  #PROJECT('cciprof.clw')
#ENDIF
```

Well, that's it. Now my dlltutor application is running with a beautiful trace window included. I've also tried these classes in a couple of other multi-DLL projects without problems. However, if you find any trouble with these modifications, please let me know.

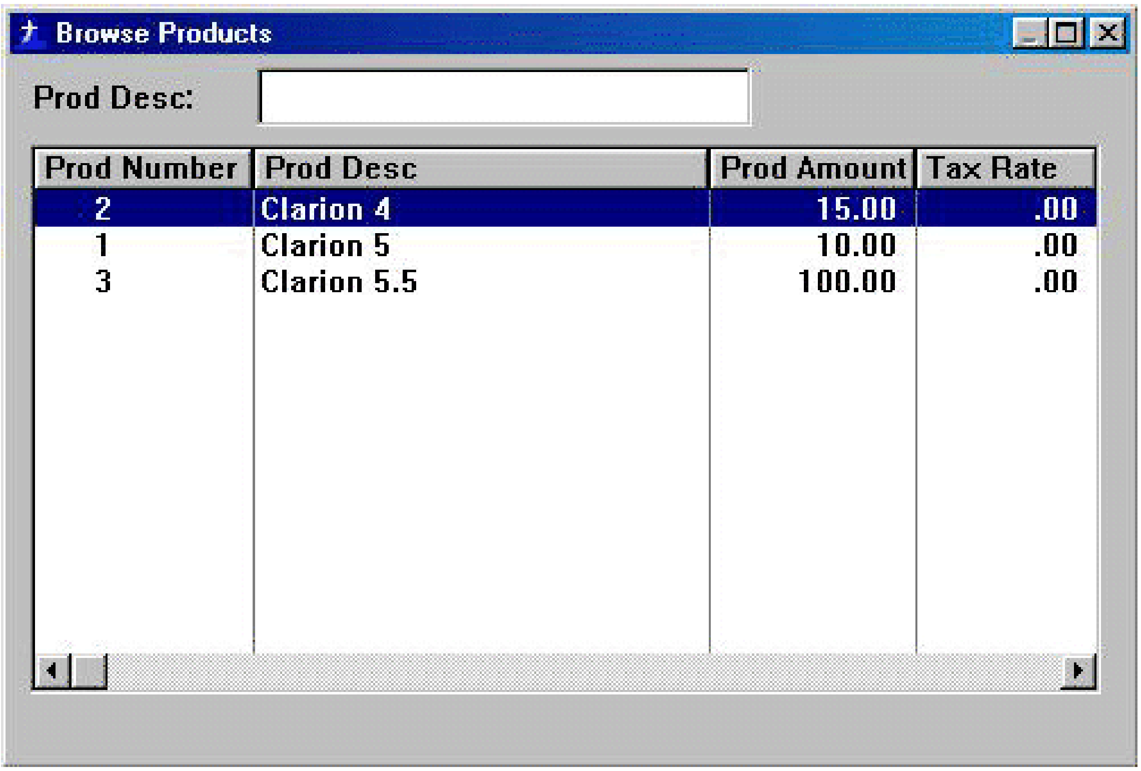
A browse fix

Another modification that I've made to this class has nothing to do with DLLs, but was implemented to avoid an annoying bug that I found when I opened any browse in an application when the trace option was activated. The last record (or the first one if I was reading backwards) was repeated several times, as shown in Figures 1 and 2.



Prod Number	Prod Desc	Prod Amount	Tax Rate
2	Clarion 4	15.00	.00
1	Clarion 5	10.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00
3	Clarion 5.5	100.00	.00

Figure 1: A browse before updating the cciProfilerClass.



Prod Number	Prod Desc	Prod Amount	Tax Rate
2	Clarion 4	15.00	.00
1	Clarion 5	10.00	.00
3	Clarion 5.5	100.00	.00

Figure 2: The same browse with the cciProfilerClass updated.

This happens because the ViewManager class keeps reading the next record in sequence until the ERRORCODE() procedure returns a non-zero value. In normal situations this only occurs when the end of file (or the begining) is reached, but if you set the Profile define on (and that's what the cciProfilerClass does) the error code could be cleared by any statement placed in the EnterProc or LeaveProc procedures. In cciProfilerClass, for example, the error code is always cleared by the statements used to access the trace queue.

I've modified the ProcStart method of the cciProfilerClass to store the error code at the beginning of the method code, and set it back at the end. It was very easy to store the original error code by using the ERRORCODE() procedure, but unfortunately there is no SETERRORCODE procedure defined in the Clarion language.

After searching around for a while, I've found that there was an internal procedure defined in the Clarion runtime library called Cla$seterror, and guess what? Yes! That procedure receives a numeric error code, and sets the internal Clarion error code to that parameter. The only thing that I had to do was to implement a new procedure mapped to the Cla$seterror procedure, and use it to set the error code at the end of the ProcStart method. Take a look at these three updates to the cciprof.clw file:

Inside the map statement:

```
module('')
  SetErrorCode(LONG ErrorCode),NAME('Cla$seterror')
end
```

At the beginning of the ProcStart method:

```
SavedErrorCode long
   code
   SavedErrorCode = ERRORCODE()
```

At the end of the ProcStart method:

```
IF SavedErrorCode THEN SetErrorCode(SavedErrorCode).
```

That's all for now. I hope you find these updates useful. As Dave has already said, the profiler/debugger classes are great tools for debugging, or when you just want to know what's really happening inside ABC.

Download the source

Matías Flores has been using Clarion since 1993. In 1999 he joined one of the SoftVelocity distributors in LatinAmerica and has been working as a Clarion instructor for the last two years. In January 2002 he left his job with the distributor to start a company called SoftMasters with his partner Juan Domingo Herrera. In his spare time Matías likes to play soccer.

Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > COM/OLE > COM/OLE

Look Out, It's Outlook! (Part 2)

by Mark Riffey

Published 2002-05-14

In Part 1 of this series I found the so-called key to the treasure for dealing with Outlook: I figured out how to use the MS Word object browser to easily find Outlook object properties and methods. This week I'll show how to put data into an Outlook database, from a Clarion applications.

Once again, I go back to MS Word's VB editor and open the Object Browser. After I add the Outlook object library to my project list, I want to take a look at the Item object and see what fun I can have with it.

To find the methods used to find and manipulate Outlook objects, I looked at the Items object in the object browser. There I found these methods.

- `.Add`
- `.GetNext`
- `.Find`

and so on. Sound familiar? Sure they do. They do just what you would expect based on their names.

To add an item, you call the `.Add` method, then you fill in the properties, then you call the `.Save` method. Anyone familiar with VB has done this many times, as it is how databases are handled as well. The code in Listing 2 uses this approach to store an appointment into Outlook.

Listing 2. Storing an appointment in Outlook

```
!To turn on Clarion's OLE error messages use = 1 below.
?ole{prop:reportexception} = 0
MyOLE = ?OLE{prop:create}
```

```
loc:status = 'Opening Outlook name space'
myNameSpace = ?OLE{'GetNameSpace("MAPI")'}
loc:status = 'Opening Outlook Calendar folder'
myApptFolder = ?OLE{clip(myNameSpace) |
  & '.GetDefaultFolder(9)'}
loc:status = |
  'Outlook Calendar folder was successfully opened'
myAppt = ?OLE{clip(myApptFolder) & '.Items.Add'}
! this method tells Outlook I want
! to ADD an item to this folder.
?OLE{clip(myAppt) & '.subject'} = clip(aty:NAME) |
  & ' with ' & clip(APL:First_Nme) |
  & ' ' & clip(APL:Last_Nme)
?OLE{clip(myAppt) & '.body'} = APL:Notes
!?OLE{clip(myAppt) & '.duration'} |
  = (APL:End_Time - APL:Time) * 6000
!MESSAGE ('"' & FORMAT(apl:date,@D02) & ' ' |
  & FORMAT(APL:END_TIME,@T01) & '"' & ' DUR=' |
  & (APL:End_Time - APL:Time) * 6000)
?OLE{clip(myAppt) & '.reminderplaysound'} = -1
?OLE{clip(myAppt) & '.reminderset'} = -1
?OLE{clip(myAppt) & '.start'} = '"' |
  & FORMAT(APL:date,@D02) & ' ' |
  & FORMAT(APL:TIME,@T1) & '"'
?OLE{clip(myAppt) & '.end'} = '"' |
  & FORMAT(APL:date,@D02) & ' ' |
  & FORMAT(APL:END_TIME,@T1) & '"'
?OLE{clip(myAppt) & '.Save'}
! this method saves the stuff I added.
myAppt{prop:release}
myAppt{prop:deactivate}
myApptFolder{prop:release}
myApptFolder{prop:deactivate}
MyNameSpace{prop:release}
MyNameSpace{prop:deactivate}
myOLE{prop:release}
myOLE{prop:deactivate}
?OLE{clip(myAppt) & '.reminderplaysound'} = -1
?OLE{clip(myAppt) & '.reminderset'} = -1
```

The two items above are important if I want to set reminders for the appointment, as well as when I want to nag the Outlook user by playing a sound. By the way, "-1" means true to Outlook just as it does to VB, so I am saying "Yes, I want a reminder and yes, I want you to nag me about it with a sound as well as a message". Isn't nagging fun?

I had trouble setting `.reminderdate`. If I coded the setting of `.reminderdate` before the due date, I got weird results. If I coded the setting for .reminderdate in after the due date, my date request was ignored and the .reminderdate value was set to the due date. I believe this may be because I did so after setting .reminderset to true, but I have not investigated this. I'll leave that as an exercise for the reader.

Before I move on, I should make note of the value I passed to GetDefaultFolder. It was 9, which refers to the calendar. This information is found in the MS Word object browser:

```
myApptFolder = ?OLE{clip(myNameSpace) |
  & '.GetDefaultFolder(9)'}
```

The code in Listing 3 stores a task (aka ToDo item) into Outlook.

Listring 3. Storing a task in Outlook.

```
!To turn on Clarion's OLE error messages use = 1 below.
?ole{prop:reportexception} = 0
MyOLE = ?OLE{prop:create}
myNameSpace = ?OLE{'GetNameSpace("MAPI")'}
loc:status = 'Opening Outlook name space'
!display
loc:status = 'Opening Outlook Tasks folder'
!display
myContactFolder = ?OLE{clip(myNameSpace) |
  & '.GetDefaultFolder(13)'}
loc:status = 'Outlook Tasks folder was successfully opened'
!display
myContact = ?OLE{clip(myContactFolder) |
  & '.Items.Find("[BillingInformation] = ' |
  & tas:sysid & '")'}
if clip(?Ole{clip(myContact) |
  & '.BillingInformation'}) = 'GetPropertyFailed'
  !"GetPropertyFailed means "not found" or
  ! "your code doesn't work" so lets add contact
   myContact = ?OLE{clip(myContactFolder) & '.Items.Add'}
else
  if clip(tas:sysid) = clip(?Ole{clip(myContact) |
    & '.BillingInformation'})
    ! last name and sysid match, so lets update instead
  else
    !add this contact
    myContact = ?OLE{clip(myContactFolder) & '.Items.Add'}
  end
end!case
?OLE{clip(myContact) & Subject} = '(' |
  & clip(tas:action) & ') ' & TAS:Task
if TAS:DueDate > 0
  ?OLE{clip(myContact) & '.duedate'} |
    = format(TAS:DueDate,@d02)
  ?OLE{clip(myContact) & '.reminderplaysound'} = -1
  ?OLE{clip(myContact) & '.reminderset'} = -1
  !?OLE{clip(myContact) & '.reminderdate'} |
    = format(tas:duedate - 2,@d02)
end !if
! remind 2 days ahead of time.
if clip(TAS:todo_contact) > ' '
   ?OLE{clip(myContact) & Body} |
  =  clip(tas:comments) |
  & '<13,10><13,10>Phone info - Hm:'|
  & clip(TAS:HomePhone) & ' Wk:' |
  & clip(TAS:WorkPhone) & ' x' & clip(TAS:Extension) |
  & ' Cell:' & clip(TAS:Cellphone) & ' Fax:' |
  & clip(TAS:Faxphone) |
  & ' email:' & clip(TAS:Email)
else
   ?OLE{clip(myContact) & Body} = tas:comments
end!if
?OLE{clip(myContact) & '.ContactNames'} |
  = clip(tas:whotalkedto) & ' ' |
  & tas:todo_contact
if clip(TAS:Importance) > ''
  case clip(tas:importance)
    of 1 to 3
      ?OLE{clip(myContact) & '.importance'} = 2
    of 4 to 7
      ?OLE{clip(myContact) & Importance} = 1
    of 8 to 10
      ?OLE{clip(myContact) & Importance} = 0
  end!case
else
   ?OLE{clip(myContact) & Importance} = 1
end!if
?OLE{clip(mycontact) & '.Save'}
myContact{prop:release}
myContact{prop:deactivate}
myContactFolder{prop:release}
myContactFolder{prop:deactivate}
MyNameSpace{prop:release}
MyNameSpace{prop:deactivate}
MyOLE{prop:release}
MyOLE{prop:deactivate}
```

In Outlook, Importance is high = 2, normal = 1, low = 0. I found this in the object browser too. I also found that the reminder date worked a little differently here than it did in the calendar folder. Odd, but why fight it?

Note that I used my sysid in a rarely used Outlook field so that I know I put this task into Outlook. There is also an EntryID property on each item that is basically the same as a sysid in our lingo. You might add a field to your file for EntryID so that your sync is a little stronger. I'll be doing that when I make my next big dictionary change so that I can have more accurate deletes in my databases after finding that I am "missing" an entry in Outlook (ie: it was deleted from the PDA).

Finding data

The Find method is a little more work, since you have to decipher what it wants for a search argument. The following code looks for a Task that has the value of tas:sysid in the billing information field:

```
myTask = ?OLE{clip(myTaskFolder) |
   & '.Items.Find("[BillingInformation] |
   = ' & tas:sysid & '")'}
```

You can invoke the `.Find` method on any property.

Summary

The best resource I've found for using Microsoft COM objects isn't MSDN, it's Microsoft Word. With a little digging in the MS Word object browser, you can probably find all you need to know about using these objects, in conjunction with the Outlook example code in this article.

Admittedly, the ideal situation would be to have a class for this interface, and perhaps I should have written one. But I can't have all the fun - you can write an article about that!

So, now you have the keys and a map to dig for treasure in Outlook. Enjoy!

Download the source

Mark Riffey has worked in the software industry, primarily in development and technical support for two internationally known enterprise software vendors, the world's premier information systems services company, a Fortune 100 manufacturer, and now Granite Bear Development. His business philosophy is simple: Be fair to your customers and yourself, surround yourself with brilliant people, work hard, be a good listener and have a little fun. Mark and his wife Jacki have two boys, Alex and Jonathan. Mark's other interests include Boy/Cub Scouting, backpacking/hiking and almost anything else outdoors, classic blues guitar, golf and photography.

Reader Comments

Add a comment

Clarion Magazine

Data Structures and Algorithms Part II - The List

by Alison Neal

Published 2002-05-17

How many times have you been asked to write a list? Lists are everywhere: there are shopping lists, to do lists, lists of people's contact details, and lists of appointments. Invoice items are fundamentally a list of items, parameters passed to procedures are basically a list, as are field definitions in a table.

There is little that people do without using lists. The list is so pervasive and so valuable that many people have spent many years discovering how to effectively represent this structure in computer memory with the greatest efficiency. Clarion developers usually use `QUEUE`s for lists, but as I said in Part 1, too many of us don't know about structures other than the `QUEUE`.

In computer science, there are several different types of list:

- Unordered Lists
- Ordered Lists
- Dummy header Lists
- Linked Circular Lists
- Doubly Linked Circular Lists
- Multi-linked Lists

I can't do justice to all those list types in one short article, so for now I'll focus on one of the most important considerations in managing lists: efficiency. I will also show some simple ways of sorting unordered lists through the use of the Quick Sort and Merge Sort algorithms.

The list

The simplest of the list types, and therefore my starting place, is the Unordered List, or simply the list.

Consider a shopping list. This is normally unordered, unless you're like me and rewrite it every week in the order that everything is stacked on the shelves at the supermarket! I will assume, however, that you are not like me, and you write your shopping list in the order that things come mind.

Now what do I do with this list? I add items to it, I read (examine or analyse) it, and I remove items from it (strike them through) when I've found them. If I want to make an attempt at ordering the list without rewriting it, I start drawing arrows all over it, pointing to where an item ought to be. My arrow usually indicates that a given item should be placed between two other items. Ideally I'd like to be able to add something before another item, or add something after another item. This action does not, however, provide what is known as an "ordered" list, as there are no clearly defined rules of precedence involved.

Sometimes I make a spelling mistake, and I have to rewrite the item. Or I know that the supermarket doesn't stock a desired item, so I have to replace it with something else that is suitable. I also want to count up the number of items in the list. This gives me some idea as to whether I need to take the SUV or the MX5. This is a very important detail because I don't want to go get my shopping and then find it won't all fit in the car.

As for a To Do list, I want to be able to deal with the different items on that list without having to rewrite it every time, and I want to be able to deal with the items out of order. I don't always want to do the first thing on my To Do list; in fact, I'd really rather leave that one until the boss has forgotten about it. I'd rather start somewhere in the middle of my list, and I don't want to be losing my place all the time. If, like me, you generally have a ToDo list that stretches for a couple of pages, you don't want to waste half your time searching for the item you just completed only 10 minutes ago.

Here is the declaration for a list:

```
ListNode    CLASS,TYPE
nextNode    &listNode
nodeVal     ULONG
            END

head        &listNode
tail        &listNode
curr        &listNode
count       ULONG
```

Does this look familiar? It's very similar to the stack covered in my last article. The stack, like the `Queue`, can be derived from the list. However, the list has many more operations associated with it. As you can see from the declaration above, to do all the things I want to do with my list, I have to keep track of the list head, tail and the item I'm currently looking at (curr). I also want to keep a count of how many items are in my list.

Like the stack, the listNode structure contains a ULONG for the value, and a reference variable to point to the next listNode. Also like the stack, the List can be implemented by either using an array, or dynamically, as shown. The dynamic structure is also known as a Linked List. Reference variables give me the links in the chain I'm going to build.

The methods required for the list are:

Method	Description
Init	Initialise the list
IsEmpty	Check whether the list is empty
LLength	Return the current number of nodes in the list
Current	Return the current position
addBefore	Insert a new node before the current node
addAfter	Insert a new node after the current node
Del	Remove the current node
makeCurrent	Make position p (passed value) current
FirstPos	Make the first node of the list current
LastPos	Make the last node in the list current
NextPos	Make the following node the current node
PrevPos	Make the previous node the current node
LItem	Return the item in the current position
Find	Return the position of s (passed value) in the list
LReplace	Replace the item in the current node
Kill	Destroy the list

The init method is simple enough; it is very similar to the init for the stack, only there are more variables to initialize:

```
head &= NULL
tail &= NULL
curr &= NULL
count = 0
```

Like the stack, the NULL value says where to stop processing when you need to iterate or recurse through the structure.

The isEmpty() function is slightly different however as there is a count being kept of the quantity of nodes in the list. The count can be used to see whether the list is empty or not rather than checking for the NULL value:

```
RETURN CHOOSE(count = 0, TRUE, FALSE)
```

If the node count is zero this function returns true (the list is empty); otherwise it returns false.

As the count variable is being maintained continuously, the function to provide the lLength() of the list only requires one line of code as well:

```
RETURN count
```

Maintaining the count of nodes continuously saves time because you don't have to recurse through the structure counting every node until reaching the NULL terminator. When a node is added the count variable needs to be incremented, and with the removal of a node, the count variable needs to be decremented. To add a node to the list, I do the following:

```
AddBefore    (ULONG yourVal)    !Add Before the current Node
t         &listNode
CODE
    t &= curr
    IF (~curr &= NULL)
        IF (curr &= head)
            curr &= NULL
        ELSE
            curr &= head
            LOOP WHILE (~curr.nextNode &= t)
                curr &= curr.NextNode
            END
        END
    END
    AddAfter(yourVal)
```

The first time I saw this piece of code it was in C, and at the time I thought I might as well give up now. It took a long time for me to understand this. So, if this looks like Double Dutch to you don't worry too much about it. Just take your time and play with the downloadable code at the end of the article.

What's going on here? First, t is set up as a temporary variable to reference the current position. Then a check is performed to see that the current node is not NULL. If the current node is NULL, then it is unlikely that there is anything in the list, so the addAfter is immediately called. However, if the current node is not NULL then there must be something in the list already, so a check is then performed to see if the current node is the very first node (the head) of the list. If the current node is the head of the list then the current node becomes `NULL` - this assumes that if something is to be added before the first node, or the list is empty, then the new node should be added to the top of the list.

Otherwise I loop through the list and find the position required and then call the `addAfter()` procedure. The loop in the addBefore() method can be confusing; it searches the entire list until it finds in the `nextNode` the value that current was holding when the procedure was entered, and which the temporary variable `t` now holds. Thus, the loop makes current the previous node to the one required, and so makes it possible to now add the new value "after" the current node.

```
AddAfter            (ULONG YourVal)
n   &listNode
t   &listNode
    CODE
    IF (curr &= NULL)
        t &= head
        head &= NEW(listNode)
?       ASSERT(~head &= NULL)
        head.nodeVal = YourVal
        head.nextNode &= t
        curr &= head
    ELSE
        t &= curr
        n &= curr.nextNode
        curr &= NEW(listNode)
?       ASSERT(~curr &= NULL)
        curr.nodeVal = YourVal
        t.nextNode &= curr
        curr.nextNode &= n
    END
    IF count = 0 OR t &= tail THEN tail &= curr.
    count += 1
```

If the current node on entering `AddAfter()` is `NULL` then the new node is to become the new head of the list. Thus a new node is allocated, then the memory allocation is checked by the `ASSERT` statement. If the current node is not `NULL` the new node becomes the one "after" the current node (`curr.nextNode`); otherwise the new node is made to be the current node.

If the list was previously empty, or the current node is the last node in the list, the new node becomes the tail. The new node is then made current and the node count is incremented.

Now to delete the current node from the list, I do the following:

```
del                ()
t   &listNode
    CODE
```

```
?   ASSERT(~curr &= NULL)
    t &= curr
    IF (curr &= head)
        head &= head.nextNode
        curr &= head
        IF curr &= NULL THEN tail &= NULL.
    ELSE
        curr &= head
        LOOP WHILE (~curr.nextNode &= t)
            curr &= curr.nextNode
        END
        IF t.nextNode &= NULL
            tail &= curr
            curr.nextNode &= NULL
        ELSE
            curr.nextNode &= t.nextNode
        END
    END
    DISPOSE(t)
    count -= 1
```

First there is a check to see that the list is not currently empty and that the curr variable is actually referencing something useful. The temporary variable is then made to reference exactly the same memory space as the curr variable is. Now one of two things can happen, depending on whether the current node is the head of the list or not. If `curr` is referencing the head of the list then, as is the case in removing the top node of the stack, the head is made to refer to the next node. The `curr` variable is also pointed to the next node, and the temporary variable holding the reference to the old head is disposed of.

However, if the current node is not the head of the list, I need to make sure that I don't lose the references on either side of that node. To do this I find the node preceding `t`, the node before `t`, and make it reference the node after `t`. This effectively drops `t` from the chain altogether so that it can be disposed of. However, if the node after `t` is NULL then `t` is the tail, and as the tail is about to be disposed of the tail must then be made to reference the previous node.

Keeping track of the current node becomes very important, or I might find myself deleting things I didn't want to. Here is the code for the `makeCurrent()` method:

```
MakeCurrent        (ULONG pos)
i    LONG
CODE
i = pos - 1
curr &= head
LOOP WHILE i>0 AND (~curr &= NULL) AND (~curr.nextNode &= NULL)
    i -= 1
    curr &= curr.nextNode
END
```

The `MakeCurrent` function's main purpose is to find the node in the position requested. To do this, the current variable is made to reference the head of the list and the temporary variable `i` is made to equal one less than the position. Then while the temporary variable `i` is greater than 0 and I haven't found the tail of the list, `i` is decremented and the curr node moves down the list. If `i` is greater than the number of nodes in the list, the curr variable will reference the tail of the list; otherwise it will reference the requested position.

Because the head of the list is being maintained, the code for the `firstPos()` method is nice and easy:

```
curr &= head
```

The `lastPos()` method is similar:

```
curr &= tail
```

The nextPos() method is slightly more difficult:

```
IF curr &= NULL
    curr &= head
ELSE
    curr &= curr.nextNode
END
```

This says if the current reference is `NULL`, if it is beyond the tail node, then the current variable should reference the head of the list; otherwise, the current reference should reference the next node regardless of whether it is `NULL` or not.

The `prevPos()` method however is a little more tricky. Here's the code:

```
prevPos    ()
t    &listNode
CODE
    IF (curr &= head)
        curr &= NULL
    ELSE
        t &= head
        LOOP WHILE (~t.nextNode &= curr)
           t &= t.nextNode
        END
        curr &= t
    END
```

If the current node references the head then the curr variable should reference `NULL`. Otherwise a temporary reference variable is used to search out the node that comes before the current node.

If you look at the positioning methods together, you should notice that they treat the list as though it were a circle. That is if it reaches the tail of the list it moves back to referencing the head and vice versa.

Now it's all very nice to be able to maintain the current node but sometimes you need to know where in the list you are. This is the function that the `current()` method performs:

```
current    ()
I    LONG
T    &listNode
CODE
    IF (curr &= NULL) THEN RETURN FALSE.
    I = 1
    T &= head
    LOOP WHILE (~t &= curr)
        T &= t.nextNode
        I += 1
    END
    RETURN I
```

If there's nothing in the list, or the current node doesn't reference anything useful, return false; otherwise loop through the list, incrementing a count, until you find the current node. Then return the count value.

The `lItem` method returns the value stored in the current node:

```
? ASSERT(~curr &= NULL)
  RETURN curr.nodeVal
```

As long as `curr` references something useful, the function will merely return the current node value.

Finding a specific item in a list is one of the most useful functions available. How many times have you gone around the supermarket and grabbed something off the shelf, and then spent five minutes trying to locate it on your list? There you are, standing in the middle of the aisle, scratching your head, while women with trolley rage run over your feet. Well, unless your list is immense this will not take five minutes:

```
Find    (ULONG s)
I    LONG
CODE
    I = 1
    Curr &= head
    LOOP WHILE (~curr &= NULL) AND curr.NodeVal ~= s
        Curr &= curr.nextNode
        I+=1
    END
    RETURN CHOOSE((~curr &= NULL),I,FALSE)
```

Fundamentally, this method loops through the list maintaining a count until it either reaches the end of the list or finds the value required. If the end of the list is reached then the code returns FALSE; otherwise it returns the count, which gives the position of the item required in the list.

If you want to replace the value of an item in the list, then do the following:

```
? ASSERT(~curr &= NULL)
  curr.NodeVal &= YourVal
```

This method merely assigns the newly passed value to the current node.

Like the stack, when you've finished with your list, you need to make sure that you clean up after yourself and thus decrease the chances for memory leaks. To Kill the list do the following:

```
Curr &= head
LOOP until curr &= NULL
    Del()
END
```

Summary

As I mentioned in the introduction, the list is a pervasive form of data storage, and its uses can be expounded upon ad infinitum. In this article I have only covered the unordered linked list, yet there are many other types available for use. I'll discuss some of these in the next article in this series.

Download the source

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

Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III Is Under Way!

Published 2002-05-21

Photos courtesy of Andrew Guidroz and Brad Noble

Andrew Guidroz II (left, below) and ETC III mastermind Lee White discuss Wednesday night's 'E' ticket ride to the Smoky Mountains.

Andrew reports that the chef at the hotel donated 30 lbs of corn and 12 lbs of butter to the Cajun feast.

That's Carl Barnes on the left with Skip Williams (who will not be doing his Binford presentation this year). A certain unnamed Cajun had to drive Carl around for 15 minutes so he could buy root beer and pretzels.

Bruce Johnson and Tommy "Tequila" Hebenstreit in the seminar room. Tommy is Clarion Magazine's reviews editor, back on the job after a longer-than-anticipated absence (no pressure, TT!). Bob Dobbins missed Tommy at the airport. That makes every single time TT has come to ETC, he has had to hire a cab.

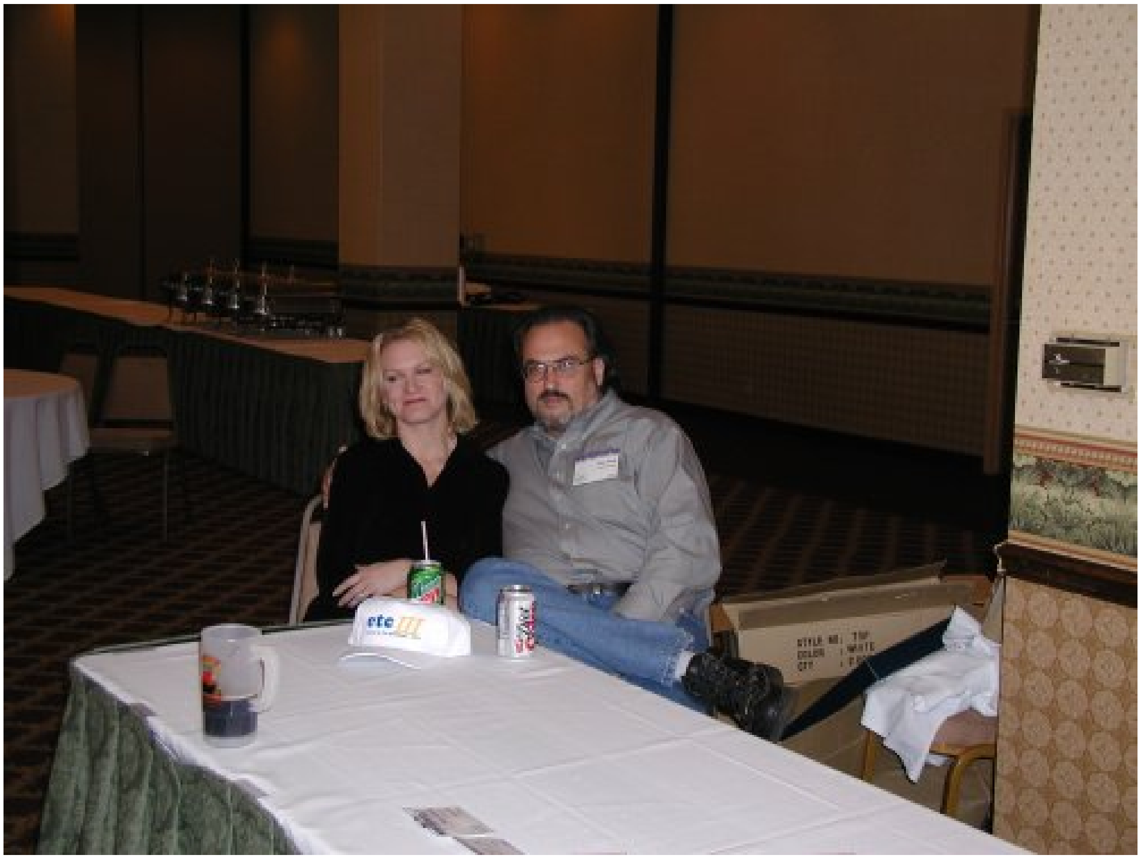
Cameraman and Team Beer linchbin Brad Noble.

Above and below, CapeSoft's Bruce Johnson giving a pre-conference seminar on NetTalk.

Lee White flanked by Earl Coker and Nik Johnson.

There are lots of new and familiar faces at ETC - see if you can spot someone you know. The weather is beautiful and cool, and the hotel lobby is packed with Clarionauts.

If you've ever wondered what that guy who posts 90% of the chat messages looks like... Note the sunglasses for disguise. Also, he appears to be staying at the "'Tater Hotel', probably for security reasons.



More pics to come, plus full reports on ETC events!

Of course, the last word on this page goes to Lee.

Reader Comments

Add a comment

Cool pictures! I wish I could be there. Friedrich
Yes, we can only envy all this company! ;-) Good luck by...
Keep them coming makes for interesting viewing putting...
Have one on me, you lucky people! I'm sorry I can't be...
Great to keep informed! Too bad I couldn't go :-)

Clarion Magazine

Home COL Archives

Topics > News > ClarionMag 2001 News

Clarion News

Published 2001-11-21

USB Print/Fax/Email From DOS Apps

If your DOS app can print to a file, you can print to a USB printer; you can fax using your PC Fax software; or you can Email your printout either as a message or an attachment.

Posted Thursday, May 30, 2002

DOS Printer

DOS Printer handles Epson character codes (bold italic, different pitches) at the character level (i.e. within a line). Written in C5b with CPCS.

Posted Thursday, May 30, 2002

EasyWinIco 1.00 Freeware

The EasyWinIco template lets you associate an animated icon with a window. This package includes the template, demo app and documentation. Requirements: Clarion 4, 5.0b or Clarion 5.5; ABC or Legacy template chains; 16 or 32-bit.

Posted Thursday, May 30, 2002

0-MeZz Bugfix

Langaard Software has made some enhancements and bugfixes to the 0-MeZz templates. A new install file is available.

Posted Thursday, May 30, 2002

gFileFind Contest Ends Friday, May 31, 2002

There are only a few days left in the gFileFind contest. You can win Gitano's Enterprise Bundle valued at $902. The inclusion of a free dBuddy and PDF Manager with any bundle purchase also ends on May 31.

Posted Thursday, May 30, 2002

Sterling Prize Drawing Deadline May 31, 2002

There are 11 prizes up for grabs in the Sterling Data prize draw. The first prize is worth up to $195, depending on which template you choose. In addition, there are 10 prizes for runners up of Product Scope 32 PRO 4.5. This excellent product includes details of hundreds of Clarion add-ons in the Clarion Third Party Profile (normally costs $49). All entrants will be notified by email of the winner's names, which will also be posted on the website.

Posted Thursday, May 30, 2002

Office Templates v2.0 Beta

SoftMasters has released the Office Templates 2.0 Beta. Notice that both the product and the documentation are beta versions - corrections and suggestions for improvement are appreciated. Because of the rewrite and translation to English, this version is not compatible with 1.0. Future releases of the Office Templates will be based on this release and will be 100% compatible with v2.0. SoftMasters is writing a couple of articles that will help you to easily upgrade from v1.0 to v2.0. These articles will be sent only to those who have subscribed to their newsletter. You can purchase any SoftMasters products with a 30% discount until May 31, 2002. Just mention the 30% ETC discount when you place the order at ClarionShop.

Posted Thursday, May 30, 2002

VideoCapture Class Wrapper Released

A demo of the Clarion wrapper for the Microsoft AVICap video capture class is now available. This app shows how to do Video Capture, Image Capture, Palette Capture, etc. in Clarion using the VideoCapture class wrapper. AVICap supports streaming video capture and single-frame capture in real time.

Posted Thursday, May 30, 2002

Das ClarioNET Upgrades

Upgrades are available for Clarion 5.5 versions of Das Security and Das Registration. Registered users can purchase these add-ons for $79.00 USD each via email to wizard@thetingroup.com. Please include the Product, Company name, Registered User Name, and credit card #, exp date, name and address. Your product will be shipped via email when order has been processed. Product Requirements: Licensed Version of CW5.5G; Licensed Copy of ClarioNET v1.1; Application must be compiled as a 32 bit - Standalone App.; Registration with Das Registration or Das Security must be current (within the last 12 months).

Posted Monday, May 27, 2002

INN Bio for 23-May-2002

This week, the Icetips News Network is pleased to present the bio of a programmer who used to be a baker, then a librarian; now, she's working on interfacing Clarion with an IBM AS/400. She makes an interesting comparison of SQL to mathematics, and has some fun photos, including a great picture of cross-country ski trails.

Posted Monday, May 27, 2002

Icetips Professional Wizards Released

The Icetips Professional Wizards are now available from Icetips Software. This product has been in development for more than six months and during the last month it has gone through eight beta testing cycles. A fully functional demo is available, limited only by the number of times you can run the Wizard Editor (40 times) and the Runtime Wizard (10 times). This product allows you to modify over 180 attributes and settings for your application before you do any generation with the wizard templates. It also allows you to register your templates in the Wizard Editor and apply them to your application and/or procedures. You can even register your own templates and use them, because all the information that is need to apply the templates is retrieved from the template files when you register them. If you buy the Professional Wizards before June 1st, you get them for US$299.00, and that includes a package from Gitano software, including their XP Icons and 12 Themes that they designed for the Icetips Professional Wizards. On June 1st, the price for the Professional Wizards will go up to US$349.00 and the Gitano package will no longer be included, but available from Gitano Software directly for US$29.95. Please note that the Icetips office will be closed from Tuesday, May 28 to June 12. Arnor and Sue will be able to check emails on semi-regular bases between May 30 and June 10.

Posted Monday, May 27, 2002

Pre-Release Clarion 5.508(h) And CLarioNET 1.2 Available

The Clarion 5.508 (h) patch is available for compatibility testing prior to general release. The patch is available both for Professional and Enterprise versions and will only work for a valid 5.507 (g) installation. The ClarioNET 1.2 patch is available for compatibility testing prior to general release. The patch requires a valid ClarioNET 1.1 installation and Clarion 5.508. In order to gain access to the patches prior to general release please reply via private email to bob.brooker@softvelocity.com requesting the download information.

Posted Monday, May 27, 2002

0-MeZz Templates Released

Langaard Software has released the 0-MeZz templates, which allow users to do mass updates similar to those done by TopScan. You can mark fields as non-updatable so no keys are overwritten.

Posted Monday, May 27, 2002

Softmasters ETC Discount

Until May 31, 2002 SoftMasters is offering a 30% discount on all its products. When you order mention the 30% ETC discount.

Posted Monday, May 27, 2002

XPMenu Lookalike Template Added

An ABC template for the XPMenu lookalike has been added - a legacy template will follow in time.

Posted Thursday, May 23, 2002

Office Templates v2.0 Available Soon

SoftMasters will be releasing version 2.0 of its Office Templates in the near future. Major changes and improvements were implemented to make the templates and classes ABC compatible. Both the classes and the documentation have also been translated into English. These classes will be distributed as freeware, just like the first release was. If you want to be notified when Office Templates v2.0 are released please visit www.softmasters.com.ar/freeware and subscribe to the newsletter.

Posted Thursday, May 23, 2002

RichReport Beta 3

Beta 3 of solid.software's RichReport is now available. New in this RTF printing engine for Clarion: Template option to load plain (unformatted) text into the RichReport control; Template option to automatically add the used memo or string field to the list of hot fields to the control template and the code template; Class bugfixes for LastField() function causing bullets to be misprinted, and restoration of report's unit setting when PrintRTF exits; Allow loading of plain, unformatted text. This is supposed to be the last beta release of RichReport. If no bugs are reported in the meantime, it will go gold at the end of the month. So this might be your last chance to buy RichReport at the reduced beta price of US$59 (gold will be US$99). Demo available; RichReport can be purchased at www.clarionshop.com.

Posted Thursday, May 23, 2002

0-HaZzle Gold

Langaard Software has released 0-HaZzle Gold. The new price is $199. You can purchase at www.clarionshop.com. Existing users can download 0-HaZzle GOLD at http://www.langaard.50megs.com/0hazzle/0hazzle.exe (install password same as the beta). The new documentation is still under development. Please remember to unregister the old template before installing the new.

Posted Thursday, May 23, 2002

PDF-Tools Evaluation Version

An evaluation version of the PDF-Tools SDK with Clarion specific Help, Templates, Classes and samples, is now available. Also includes OCX's for other dev platforms and C/C++ examples if required. PDF-Tools offers an growing list of features and requires the distribution of up to three DLLss with your apps - no printer drivers need to be installed on your equipment or your users.

Posted Thursday, May 23, 2002

SealSoft's ETC-III Special Discount

Until May 31, 2002 you can purchase any SealSoft product at a 20% discount.

Posted Thursday, May 23, 2002

Product Scope 32 PRO Version 4.5 Released

Product Scope 32 PRO Version 4.5 is used to maintain the Clarion Third Party Profile Exchange at http://www.encouragersoftware.com/profile/clarlinks.html. This is an internet bookmark and comparative shopping program. File Search - Any file type, text within files, image preview, Copy, Move, Launch. Spreadsheet and Crosstab reports available (Spreadsheet Option). Custom Views - Customize fields to display, quickly sort a single field. Database convenience for tracking product, manufacturer, and place to buy information. Serve as an alternative address book for your email program. Search All text fields including memos to find a particular Product or Vendor FAST. Reference Archived HTML Files. Associate Product Prices with International Currencies. Basic File Management for Downloading Files From The Web and Installation. Document Insight Development Web Page Captures - iCapture 2, ZUTW PRO. Associate PDF files with product record. Launch Adobe Acrobat Reader to view PDF file.

Posted Thursday, May 23, 2002

EasyAnimation 1.0 Released

EasyAnimation 1.0 lets you brighten applications with animation and sound. Easy insertion of animated clip into the window, including: AutoPlay after opening; Transparency; Centering in the Region rectangle; Play only under mouse cursor. You can specify WAV file start frames and looping. Also assign a characteristic sound (WAV) with any program event. Requires Clarion 5.0b or Clarion 5.5, ABC or Legacy, 32 bit. EasyAnimation is limited to simple AVI files. This means the EasyAnimation supports animation clips that are uncompressed or RLE8 compressed, have a color table, contain a single video stream, and have no sound (but you can synchronize a WAV file with the animation).

Posted Thursday, May 23, 2002

Application Shell May Special

You have just ten days left to take advantage of the Application Shell May Special. Save $300.00 on the purchase of the Popular Edition.

Posted Tuesday, May 21, 2002

Purchase XPMenu Online

Compad Software's XPMenu lookalike is now available for purchase through SWReg.

Posted Tuesday, May 21, 2002

Nice Touch Solutions ETC-III Special

Nice Touch Solutions is having an ETC-III special through Friday, May 31, 2002. The WizPak bundle is available now for $649. WizPak includes: Query Wizard 5; Report Wizard 1; View Wizard 1; Spreadsheet Wizard 2 (to Excel); CrossTab Wizard 2 (to Excel). Separately these products would cost $1,025. If you already have some of the products and would like to get the full WizPak, Nice Touch is providing full or partial credit of previous purchases (depending on the date of the purchase). Conference attendees also get a choice of Application or Dictionary Assistant. For an addition $59, they can both. This is to reward those supporting Lee White and his band of workers putting on this conference again (even though he vowed he would not ever do it again). Steve Stockstill and Arnold Young will be in attendance from Tuesday 5/21-Sunday 5/26/2002, so there may be a delay in processing your orders on those days.

Posted Tuesday, May 21, 2002

Linder Software Office Closed May 23-26, 2002

http://www.lindersoftware.com The Linder Software office will be closed May 23rd through May 26th. All calls, e-mails and confirmations will be returned on May 27th.

Posted Tuesday, May 21, 2002

Gitano ETC 20% Off Special

This week, May 20 to May 26, Gitano Software is offering 20% off its entire product line, including bundles which are already reduced. Plus Gitano will give you a free copy of PowerSearch.

Posted Tuesday, May 21, 2002

CapeSoft ETC Madness Specials

For the week of May 20-24 CapeSoft is having an ETC Madness Special. All CapeSoft Accessories are available on special via www.clarionshop.com as well as the CapeSoft Utilites.

Posted Tuesday, May 21, 2002

Purchases of BigTamer(tm) During ETC-III

Bo Schmitz and Earl Coker will both be attending ETC-III and will not be able to get and answer email, so unlock codes from purchases at DeveloperPLUS will be unavailable during the week of May 20th thru May 26th.

Posted Monday, May 20, 2002

Outlook Import Template Updated

BigOutlookTamer(tm) has new features added, including: a renamed listbox action display header; "Allow duplicates" checkbox for contacts; "Import only first line of address" checkbox. Demo available.

Posted Monday, May 20, 2002

Coollook Template 1.1 Available

The Coollook template adds an Outlook style (or Leg2Abc style) menu to your application. You populate a Coollook menubar just like you populate a standard Clarion menubar. Coollook template is available for C55 legacy and ABC and is still US$49 at ClarionShop. Improvements include: ; Resizable with a splitbar - can be disabled; Select default width of Coollook; Customization of the colors of the buttons, the background of Coollook; Choice of exact Outlook style or Coollook style; Automatic maximize of called procedure - an be disabled; Refresh the display of Coollook manually.

Posted Monday, May 20, 2002

Free Software From C&G

Free Clarion software for developers, from C&G: Leg2ABc, helps developers in converting applications from Legacy through ABC; WinUpx, a windows end-user interface for the DOS compression tool UPX; OneEXE, a template that provides two extension templates that allows an application to be started only once; Barcode, a template that provides a set of functions that display, resize, print and calculate checksums of EAN13 and UPC-A barcodes; Num2Words, a free simply-to-use function that converts a number in words (French version only); AppRegister, a set of templates that creates a serial number for your application, stores it in a file wherever you want, displays a customizable register-window regularly (you fix the elapsed time between two displays) until the end-user enters the right key (calculated from the serial number); TabIsEnter, which substitutes the enter key for the tab key.

Posted Monday, May 20, 2002

INN Bio For 16-May-2002

This week, the INN Bio features the last in the IceTips series of presenters for ETC. Born in England but raised all around Europe, he became one of his country's last "boy soldiers". We know him as an ex-TopSpeed employee and Clarion's guru on COM.... and as a "piper". Someday we'd love to hear him play the bagpipes (great photo of his former band). Read about him, and the latest news on the Icetips Professional Wizards (and a little other news).

Posted Monday, May 20, 2002

JT Free Progress Bar Replacement Template Updated

For those that like the slightly different JT Progress Bar, a new "Manual" control template has been added to the set. This doesn't require a process procedure or a standard progress control. Just add and size the template, give it a variable that will contain the percent complete value as your procedure progresses and it does the rest. Also included is a Logo progress template (thanks to Jim Kane for the idea). This template either stretches or unveils an image such as a logo or banner as the procedure progresses.

Posted Monday, May 20, 2002

C5.6 Info And Clarion ASP Example Applications

SoftVelocity has updated its web page covering the April technology workshop. Now available: a link to updated ASP example applications.

Posted Monday, May 20, 2002

IceTips Software Office Closed May 28 To June 12, 2002

Icetips Software will also be closed May 28 to June 12, both days included, due to a trip to Iceland. Arnor and Sue should be able to check emails from May 30th to June 10, but there will be 2-3 days at each end where they will not be in internet contact.

Posted Monday, May 20, 2002

Solace Software Holiday

Simon Burrows will be lying on a beach on an island in the middle of the Mediterranean from May 9th to 27th, so the office will be shut during this period.

Posted Monday, May 20, 2002

SealSoft's xWhatsNew v1.0

SealSoft's xWhatsNew is a template for automatic generation and viewing of a product version history file. Installs for Clarion 5.5 and Clarion 5 with example and source code available.

Posted Monday, May 20, 2002

RichReport Beta 2 Released

RichReport Beta 2 is now available from solid.software. Changes include: Added LoadFromBlob() method; Added LoadFromResource() method; Added Replace() method; The PrintRTF() method now supports single, one-and-a-half and double line spacing; Added support for CPCS Creative Reporting Tools (Thanks to Larry Teames!); Added code templates to load, print and replace RTF texts; Added options to load RTF text from BLOBs and resources (RTF files linked into your EXEs/DLLs); Added option not to load any text automatically (instead, use the new code template to do so); Added option not to print text (instead, use the new code template to do so).

Posted Wednesday, May 15, 2002

ClarioNET Deployment Manager

The ClarioNET Deployment Manager is now available. This product offers full management and true multi-server load balancing for ClarioNET enabled Clarion programs.

Posted Wednesday, May 15, 2002

Native Clarion XPMenu Lookalike

Ronald van Raaphorst's new XP side menu add-on is now available. The price is $75.

Posted Wednesday, May 15, 2002

RInstall Version 1.c (Beta) Available

A new beta (1.c) of Rinstall is now available. This version now checks for setting the computer date back to catch illegal extension of demo period. Documentation and unlock app have been updated. Use the same password you got from ClarionShop. RInstaller can be purchased for $99 from www.clarionshop.com. This price is discounted to $69 during the beta stage.

Posted Wednesday, May 15, 2002

Legacy To ABC Template Bundle Price

New from Comsoft7, BigLegacyToABCTamer(tm) converts over 77 legacy embeds to ABC embed points. It also converts Window, control, and Browser embeds. Use this template wizard to convert the currently loaded app, or select a separate TXA to use. Creates a backup and can import the TXA automatically back into the IDE. Cost is $249, but this template is available in a bundle with BigFileSyntaxSearchTamer for $189 until May 31, 2002.

Posted Monday, May 13, 2002

XPMenu Product

Ronald van Raaphorst has created a Clarion XPmenu lookalike. You can change all colours, fonts, icons etc.and even minimize the menu to the left side (Click to reopen). If there's interest, Ronald will set up a proper website, a manual, a template and sell it for $50 or something like that.

Posted Monday, May 13, 2002

File Syntax Change Template

Comsoft7's BigFileSyntaxSearchTamer(tm) template searches embedded code for keywords on the file syntax changes, and also will search any other words you add to the list. It then creates a text file with the procedure name, embed point and line of code it detected. Cost is $59. This template is available in a bundle with BigLegacyToABCTamer for $189 until May 25, 2002.

Posted Monday, May 13, 2002

INN Bio for 8-May-2002

While this week's INN Bio continues with INN's series of presenters at ETC. It is also the first bio of a SoftVelocity employee! Instructor, coder, tech support, documentation... it seems he's done it all. And it all started in the Navy, as a "cryptologic technician". See him skiing, visiting with his dad, and well-dressed in his role as a "finatic".

Posted Monday, May 13, 2002

Icetips Pro Wizards

New in the Icetips Pro Wizards: you can not just include third-party templates in your wizarded apps and procedures, but you also pre-set the template values, too. And you can create separate template "theme" files -- if you use different templates in different projects, you can create and save a different setup file (theme) for each project you're working on. Then, when you create a new procedure (or app) for that project, all the templates you normally use in that project will be included, with their property values already set however you want them.

Posted Monday, May 13, 2002

Batch Email And Save Report

VividHelp has released a new template which employs the employs C55 ABC Document Send template. This makes it absolutely independent of any particular emailing program, and let your customers with a click of button send multiple reports to as many recipients as they like, or with a click of another button save those reports to disc. Demo available.

Posted Monday, May 13, 2002

SoftMasters Controls Pack 1.0 Released

SoftMasters Controls Pack is a set of wrapper classes and templates that allows you to easily include four different common controls in your Clarion applications. This pack includes animation controls, IP address controls, progress bar controls and trackbar controls.

Posted Monday, May 13, 2002

EasyMultiTag 2.01 Released

The latest release of EasyMultiTag includes the following bug fixes: problem with using shift+mouse and filtered locator together fixed (thanks to Aimo Rintala); "Apostrophe in string" bug fixed (thanks to Benjamin Morehouse). All other known bugs have also been fixed.

Posted Monday, May 13, 2002

Windows XP Compatibility Update

This article from Microsoft details XP compatibility issues and fixes.

Posted Sunday, May 12, 2002

RichReport Beta 1.0

Beta 1 of RichReport is now available. This is a small class with surrounding templates that allow you to print RTF text in standard Clarion reports. No coding required - simply put a control template onto your report and tell it where to get the text from (can be a database column or a RTF file). Demo available for Clarion 5.5. RichReport supports Clarion 5.0 and Clarion 5.5, Legacy and ABC templates. Comes as .LIB and .DLL versions, supporting both standalone and local runtime libraries (32bit only). The price for the beta version is US$59,, or US$79 for the gold version. Available from solid.software and ClarionShop,

Posted Wednesday, May 08, 2002

VizAcc Screen Capture Beta

VizAcc Screen Capture beta release 3.0.0.0 is now available. Features include: Select Scanner;

Acquire from Scanner; Browse ThumbNails; Color Adjust; Grayscale; Black & White; Color Conversion; Selection; Average; Edges Strong; Edges Weak; Edge Enhance; Emboss Color; Emboss Light; Emboss Medium; Emboss Dark; Etch; Blur; BlurSoftly; BlurMore; BlurBartlett; BlurGaussian; Glow; LaplacianHV; LaplacianOmni; Maximum; Minimum; Median; Negative; Prewitt; Sharpen; SharpenMore; SharpenLess; SharpenDirectional; SobelPass; UnSHarpMask; TraceContour; Waggle; Pattern; Zero. Cost is US$20.

Posted Wednesday, May 08, 2002

The Wizzie Group - Call For Second Hand Wizatron Resources

A group of Clarion programmers have formed the Wizzie Group to share information and resources on Wizatrons. The aim of this group is to get a coherent set of executables, files and documentation available for download, as the currently available Wizatron tutorial files do not correlate to the Handbook tutorial.

Posted Wednesday, May 08, 2002

Duke Application Shell May Special

During the month of May Gerald W Duke and Company Inc. is offering the Application Shell Popular Edition for $795.00, a $300.00 savings. The Popular Edition includes Accounts Payable, Accounts Receivable, Check Reconciliation, General Ledger, Inventory, Order Entry, Purchase Orders, and Payroll.

Posted Friday, May 03, 2002

New SysViewPack Bundle

Now available from solid.software, the SysViewPack bundle consists of SysList and SysTree, the list-view and tree-view common control wrapper classes. The price for this bundle is US$159, a savings of $39. SysList is a wrapper class for the list-view common control, allowing you to have listboxes with the large icon, small icon, list and report view you'll know from popular applications such as the Windows Explorer, Outlook Express and many more. SysTree is a wrapper class for the tree-view common control, bringing back the real Windows look & feel to Clarion applications dealing with tree-like display of data. As an addition, this package contains a class called SysDirTree which let's you browse your computer's folders like you can do in the Explorer.

Posted Friday, May 03, 2002

twTools 1.7 Released

Version 1.7 of twTools consolidates some minor bug fixes in various areas. This release includes fully coded example apps showing how to load a DLL at runtime, call code in the DLL, and have the DLL call code back into your EXE. Demo available.

Posted Friday, May 03, 2002

INN Bio for 2-May-2002

This week's bio features the quintessential Texan, Clarion's own Cowboy. An expert in SQL, he's branched out into web work, including co-owning a web hosting company and a new business, "Paywire". A man with quite a diverse past, he now stays busy with his ranch, his family, and his work with technology... and his Harley, of course.

Posted Friday, May 03, 2002

NetTalk Goes Gold

The current version 2.0 gold release of CapeSoft's NetTalk is now available. The product has changed substantially from the early days of December 1999. What started out as a product to provide robust TCP/IP peer-to-peer communications between Clarion applications has grown into a multi-headed monster that encompasees virtually every corner of TCP/IP networking. Along the way CapeSoft added email (send/receive), DIP, news (NNTP), NetRefresh, HTTP, FTP, documentation Jump-Starts, Dial-Up-Networking support; UDP, WebBuilder support, passive FTP, a proxy example, and web and email 168 bit encryption.

Posted Friday, May 03, 2002

NetTalk At ETC

If you are going to ETC this year then make sure to catch the special NetTalk Training which will happen on Tuesday 21st from 3pm to 5pm in the main ballroom. Bruce is record the session on video. He's not promising anything fancy, but hopefully CapeSoft will be able to make the session available in a recorded format sometime later in the year.

Posted Friday, May 03, 2002

Free PowerSearch License

Do you want a free PowerSearch license? Send Gitano Software a translation for all the text in PowerSearch and you will receive a free copy of this utility. Please ask first to make sure that your particular translation hasn't already been done.

Posted Friday, May 03, 2002

KSprbar 1.20 Released

Version 1.20 of the KSprbar progress bar control template has been released. This version supports placing of text inside of a progress bar to display progress percent or other info. Demo available.

Posted Friday, May 03, 2002

Bug Poster Source Updated

Updated source code for Bug Poster is now available. This includes two separate versions (DLL and EXE).

Posted Friday, May 03, 2002

Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III: Keynote Address - Bob Zaunere

by Tom Hebenstreit

Published 2002-05-22

Two years ago, the weather at ETC II was fairly gloomy - storms, rain, thunder, lightning and flooding - just about like the future of Clarion itself. Bob Zaunere also spoke at that conference, having just formed SoftVelocity and taken over Clarion from the ailing, failing TopSpeed Corporation. How appropriate then that as Bob delivered the keynote for ETC III the sun would be shining and the outlook so rosy. New products, new features, and whole new worlds for Clarion programmers to explore.

Before I start, I want to mention that I attended the SoftVelocity Technology Workshop a month ago in Florida, where SV staff took two and a half days to go over what Bob was trying to cover in one hour. Needless to say, today he was talking a heck of a lot faster than I can type or write, and he still ended up running to more than two hours (it took me two full articles to cover the workshop as well.)

For the most part, Bob spoke on Clarion 5.6. Topics included:

- More changes and new features have been added to 5.6 than in the transition from 5.0 to 5.5 (more on this below in the questions section.)
- The new threading model that features preemptive operating system threads. This replaces the existing threading model which uses smoke and mirrors to give you the appearance of threading while actually running your entire application on only one true operating system thread.
- Threaded classes will have constructors and destructors called for each instance as it is instantiated for each thread (prior the RTL only called them when the global object was created). Every 32-bit component of Clarion has been changed to support the new threading model.

- The ability to dynamically create file, queue and group structures within your program.

- Both Clarion and ABC template chains have been improved in both large and small ways. The Clarion templates are being updated to add new features such improved Edit-In-Place and more. The ABC classes have been updated for the new threading model. New classes have been created to wrap around the lower level functions like ADO and the XML parsers.

- Form based record navigation, i.e. being able to step forwards and backwards though a file using a form format rather than a browse format. This includes locators and setting filters and sorts on the fly. You can also insert a new record right from the same form, and changes can be batched. One question brought out the fact that the procedure Bob was using to demonstrate this feature had no embed code at all - all of the features were driven by templates. Bob also pointed out that there is still a browse behind the form; you just don't see it (that's why you get those locators, sorts and filters.)

- Improvements to reports, including more control of detail breaks (this generated a spontaneous round of applause.)

- Templates for making the selected field on a form more obvious, including things like adding an arrow character to the left of the field, colorizing, and so on.

- More customization of application look-and-feel at design-time, with the ability to save themes and choose them again when going through wizards.

- The ability to specify business rules within the dictionary. Basically, you can use anything that will work with the EVALUATE statement.

- Client side triggers in the TopSpeed drivers, where you can have the driver call back to you on inserts, updates and deletes.

- Additional SQL support for stored procedures at design time. From within template prompts in the IDE, Bob called up stored procedure prototypes from the database. It was easy to see the procedure parameters, data types, and so forth.

- ADO templates and hence more explicit SQL support. Both ABC and Clarion template chains will be supported.

- XML support either using ADO (if your database supports it), Microsoft libraries that ship with Internet Explorer/Windows, or via an open source library. Each method has its strengths and weaknesses, but the key point is that you will have multiple options to choose from.

- They are planning on having a lot of online training for the new features and technologies.
- And a whole lot more...

Bob demonstrated the new threading, along with some of the code. He did point out that with the new power, there come a few more things to be aware of. For example, you can suspend and resume threads now, and if you happened to suspend the active, currently running thread without having a method to wake it up, you will have quite effectively hung up your program!

Bob made an interesting point that ADO can be implemented selectively within an application that otherwise uses the native drivers. For example, with ADO you can specify the SQL cursor type whereas the native drivers choose the type for you. He described a situation where, by using ADO and controlling the type of cursor in a few heavily used procedures, they vastly reduced the amount of network traffic their application was creating when scrolling through a dataset.

Here are a few of the more general questions that were asked, along with the gist of Bob's answers:

> **Will 5.6 include the new ADO templates?** That depends which edition you own. They will be in the box for Enterprise, and probably will be an add-on that can be purchased for Professional.
>
> **With all the changes, why isn't this 6.0?** It is not a 32-bit IDE, and he feels that there had been too many promises made by TopSpeed, etc., that Clarion 6 would have the feature. So Clarion 6 will be 32-bit, but you'll get 5.6 first.
>
> **So why isn't it 32-bit?** SV already has much of the system done. What they are working on is not just converting the IDE, but redesigning selected sections to not only add functionality, but to accommodate SV's longer term goal of generating multiple languages. For example, it doesn't help much to convert the editor if that editor is specific to just one language.
>
> **Does 5.6 mean that 6.0 is four more releases away?** No.
>
> **Price?** About the same as the upgrade from 5.0 to 5.5.
>
> And inevitably after hearing about and seeing the new features, **when will 5.6 be**

available? I quote: "As soon as possible, as soon as it is done."

Will there be a general public or paid beta? No. Bob felt that it was counter-productive in many ways. They will be seeking qualified, active beta testers rather than just throwing the code out to the wolves like TopSpeed did.

Are the Wizatrons gone? To quote Bob: "I think the Wizatrons will remain in our memories forever."

What about 16-bit support? It will remain in the product, but it will have no more enhancements. They are basically culling out the old 16-bit code so that they can concentrate on what is now the common standard, i.e., 32-bit.

Marketing? He said they are doing well, and more marketing will be coming as they make the tool more compatible with existing technologies.

How big is SV? He said that at any given time they may have as many or more developers working on various projects then TopSpeed had. The key is that they contract out research projects etc. as needed.

What is being phased out? Other than 16-bit, really nothing at the moment. Bob would like to go to pure dot-syntax and move away from the prefix:colon type syntax, but that was just a sort of generalized wish.

Summing up, Bob emphasized SoftVelocity's goal of making Clarion more compatible with existing standards and technologies such as SQL, COM, ADO and XML.

Whew!

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

Reader Comments

Add a comment

Mark Riffey adds this comment: Regarding Clarion/ASP - Bob...
Tom, you did a great job summarizing what was really a 2.5...

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III: CapeSoft NetTalk - Bruce Johnson

by Tom Hebenstreit

Published 2002-05-23

ETC this year is featuring a group of extra sessions, hosted by vendors of various third party products. Kicking it all off was a two-hour talk given by Bruce Johnson of CapeSoft that focused on their NetTalk TCP/IP communications accessory.

Oriented more as a training session than a sales talk (about 60% of the people attending the talk were already NetTalk owners), Bruce started by going over some essential terminology, including TCP/IP, the networking protocol that underlies the Internet (and quite possibly your own local LAN as well), sockets, and ports. He then described the three primary functional areas of NetTalk, which are:

NetSimple

This is a collection of objects that take care of the basic nitty-gritty of establishing and maintaining Internet-type connections between computers, either locally or across the Internet. NetSimple supports UDP as well as TCP. Of the two methods, picture TCP as being like a telephone conversation, i.e., a two-way connection. It ensures that each packet is sent and received correctly, and in sequence (on a per connection basis). UDP, on the other hand, is connectionless protocol, more like a radio broadcast - it has no way of knowing whether any intended recipient is listening (and thus has no error checking.)

In addition to the basics, NetSimple includes the most common types of Internet functionality that you might want to add to your programs, things like SMTP (sending email), POP3 (receiving email), FTP (transferring files between machines), HTTP (communicating with web servers) and NNTP (posting to and reading from newsgroups). Apart from those built-in items, NetSimple can basically be used to talk to anything that speaks TCP/IP (networking hardware, for example). Note that while NetTalk can talk to a web server, it is not a web browser. You can download a page, but all you'll get is the pure HTML. You would still need some type of HTML renderer (i.e., a browser) in order to view it as it would appear on the web.

The NetTalk Protocol

The second major area Bruce covered was using the NetTalk protocol. This is a proprietary networking protocol developed by CapeSoft, although it still uses TCP/IP, sockets and ports underneath to actually move packets around.

Bruce pointed out that the NetTalk protocol is best suited to applications that run on a LAN, primarily because it can have problems with firewalls and proxy servers. The reason for this is that the NetTalk protocol establishes connections in both directions, e.g., your machine connects to another machine and then that one connects back to yours. Since the primary purpose of a firewall is to block incoming connections from outside the local LAN, this prevents the full NetTalk protocol dialog from starting with a machine outside your LAN. While there are a few ways to get around this, CapeSoft is working on revamping some of the NetTalk protocol to make it more compatible with firewalls. Bruce said that their changes will be 'under the covers' and won't affect the way you interact with it at the object level.

The NetTalk protocol was developed with a focus on robustness and simplicity rather than speed (this doesn't mean it is slow, but that speed was not a primary design goal). It is not IP dependent, and ensures that packets are sent/received in order, it can compress/encrypt packets and does automatic connections. That last item is key to the NetTalk protocol - all running NetTalk servers of the type of service your program wants to access are always automatically available and connected as soon as your NetTalk protocol program starts. If you wish, though, you can also make a NetTalk server private, i.e., it won't announce to the world when it is running.

An interesting point that Bruce made was that within the NetTalk object hierarchy a client is derived from a server, so clients can not only talk to servers, they can talk to other clients (peer-to-peer).

Like NetSimple, the NetTalk protocol objects also have quite a few built-in services (functions) such as chat, file transfer, network refresh and more. Even if you don't want to use a lot of the extras, NetTalk can, for example, be used to do basic administrative tasks like remotely shutting down your programs. Where is that useful? One example Bruce gave was trying to update a multi-user program -NetTalk would let the system administrator ensure that all users were out of the program when needed.

Throughout the presentation, Bruce used the demo application included with NetTalk to show various features working, and then showed the embed code behind it (very little for most of the built-in features.)

Coding with NetTalk consisted primarily of using two or three embed points within the NetTalk classes. Ninety percent of the work is done is in a method called 'Process', which is called by NetTalk every time it has something new to tell you (i.e., every time there is an incoming event.) Basically, NetTalk is automatically handling all of the details of the connection the other machine, but you still have to decide what to say during your conversation. When using NetSimple, for example, you would check the packet type (provided by NetSimple) and perform the appropriate action. If the type was a new connection, you would do whatever you want to handle that type of transaction.

All in all, NetTalk seemed easy to use, but very deep.

Dial-Up Networking (DUN)

The third major functional area of NetTalk is interfacing with Dial-Up Networking, the tool that Microsoft provides to take care of actually connecting your machine to another network (like when you use your modem to access the Internet). The NetTalk DUN support lets you create, read, update and remove DUN connections from within your programs. It also handles initiating dial-up connections and hanging up.

Watch for a full review of NetTalk in an upcoming issue of Clarion Magazine.

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

Reader Comments

Add a comment

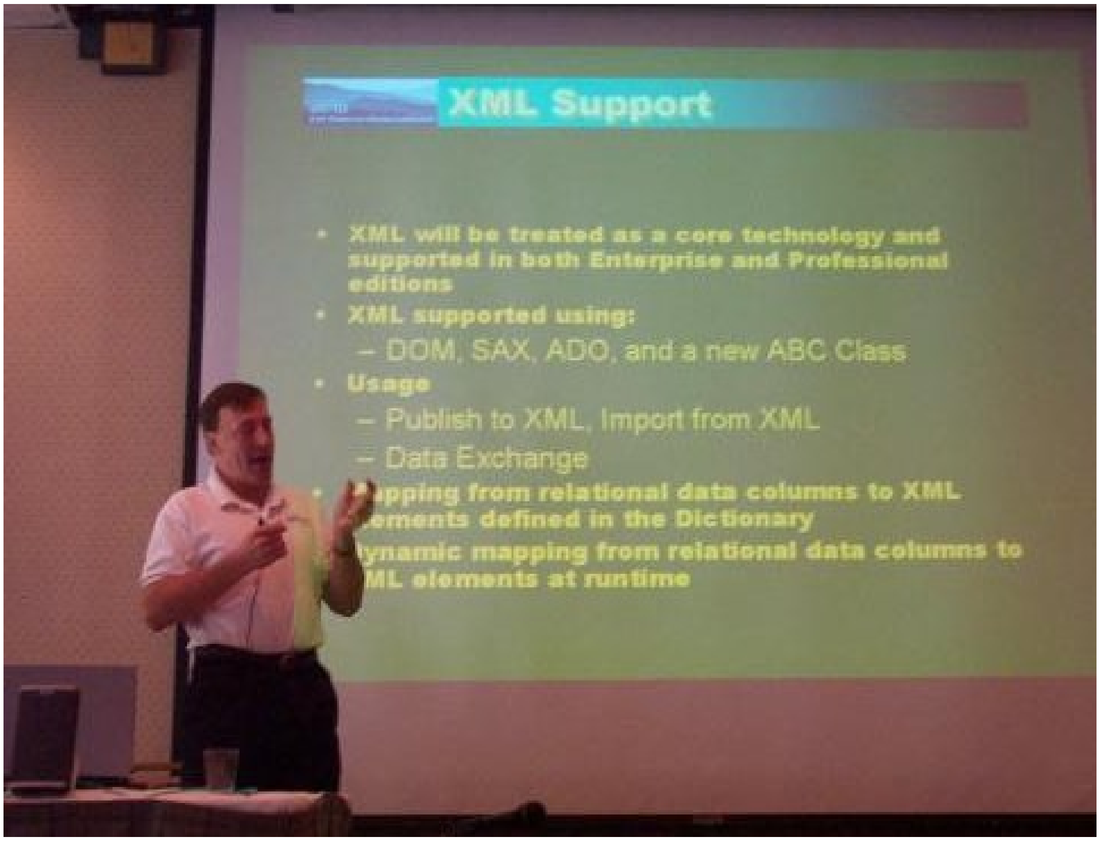
Clarion Magazine

ETC III: More Pics, Including The Cajun Feast

Published 2002-05-23

Photos courtesy of Mark Riffey

Brad Noble signing in Ben Greene and Jim McCormick at ETC registration gets underway.

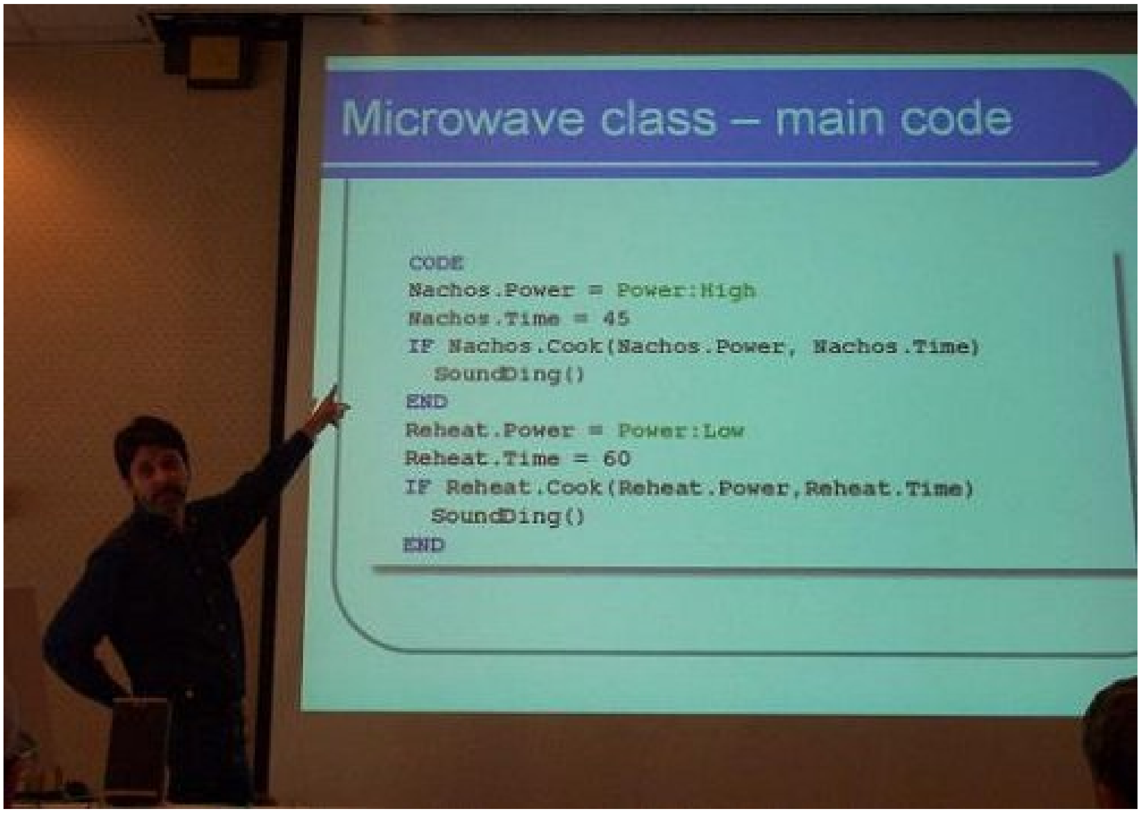


XML Support

- XML will be treated as a core technology and supported in both Enterprise and Professional editions
- XML supported using:
  - DOM, SAX, ADO, and a new ABC Class
- Usage
  - Publish to XML, Import from XML
  - Data Exchange
- pping from relational data columns to XML ements defined in the Dictionary
- ynamic mapping from relational data columns to ML elements at runtime

Bob Z entertained the troops with his air guitar, plus some very interesting information on upcoming SV products.

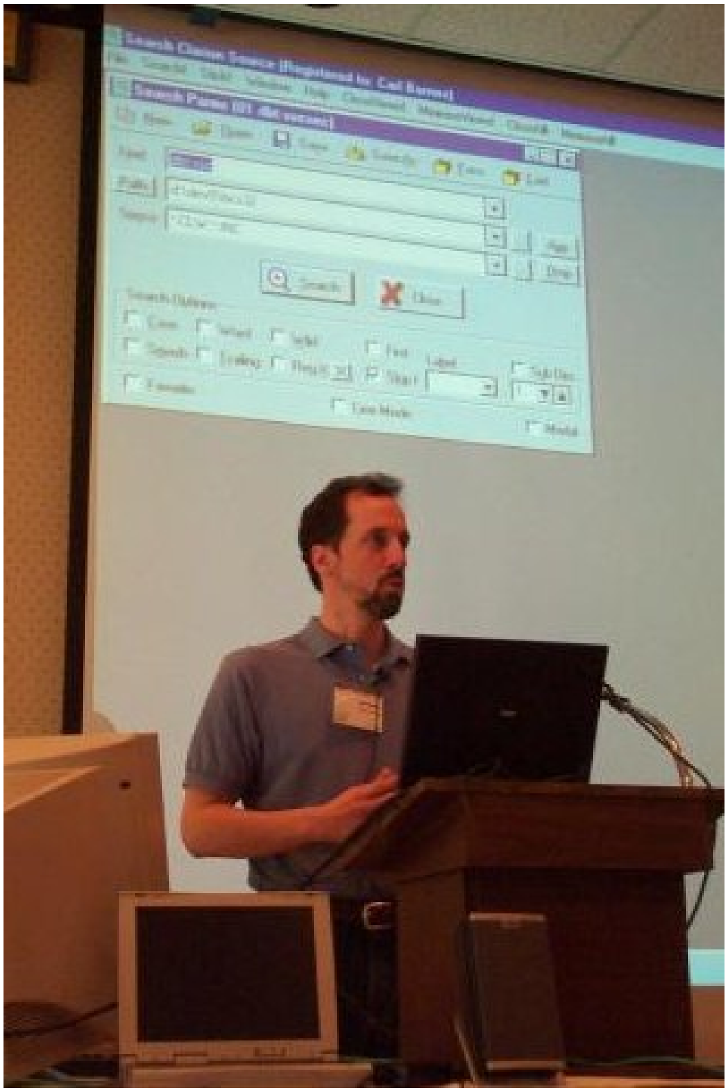
From left to right, Andy Ireland, famed Clarion COM developer and someday pub owner; Russ Eggen, former TopSpeed Clarion trainer and, ah, *occasional* contributor to the newsgroups; Michael Brooks, creator of ClarioNET; and South Africa's Nardus Swanevelder, who won his ETC conference registration in the Clarion Magazine Sweeps!



Microwave class – main code

```
CODE
Nachos.Power = Power:High
Nachos.Time = 45
IF Nachos.Cook(Nachos.Power, Nachos.Time)
  SoundDing()
END
Reheat.Power = Power:Low
Reheat.Time = 60
IF Reheat.Cook(Reheat.Power,Reheat.Time)
  SoundDing()
END
```

Russ Eggen, proving that one of the main themes of ETC really is food.



Carl Barnes demonstrating his invaluable Clarion Source Search utility. Carl discovered his inner salesman a few DevCons back.

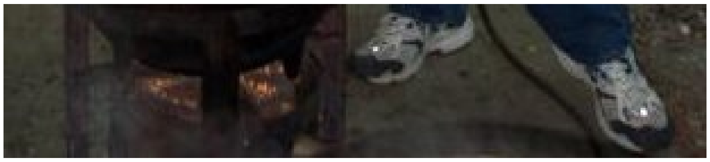
And now, it's time for the Cajun feast in the Smoky Mountains...

You wouldn't normally associate Cajun culture with the Smoky Mountains of Tennessee. But not far from Gatlinburg, tucked away in a forested glen, is a cooking shelter by a stream. Every two years since 1998 that's where Clarionites gather to eat a feast prepared by Andrew and Sabrina Guidroz, this year assisted by Andrew's son Joseph. And a feast it is!

Transportation to the site is exclusively by Gatlinburg Trolley, unless you happen to be one of the cooks, who begin preparations early in the morning.

Sabrina and Joseph hard at work. Well, Joseph's hard at work - Sabrina's having way too much fun.

Sabrina and Andrew in the background, with ETC's Bob Dobbins going for a second helping of Lee's corn bread.

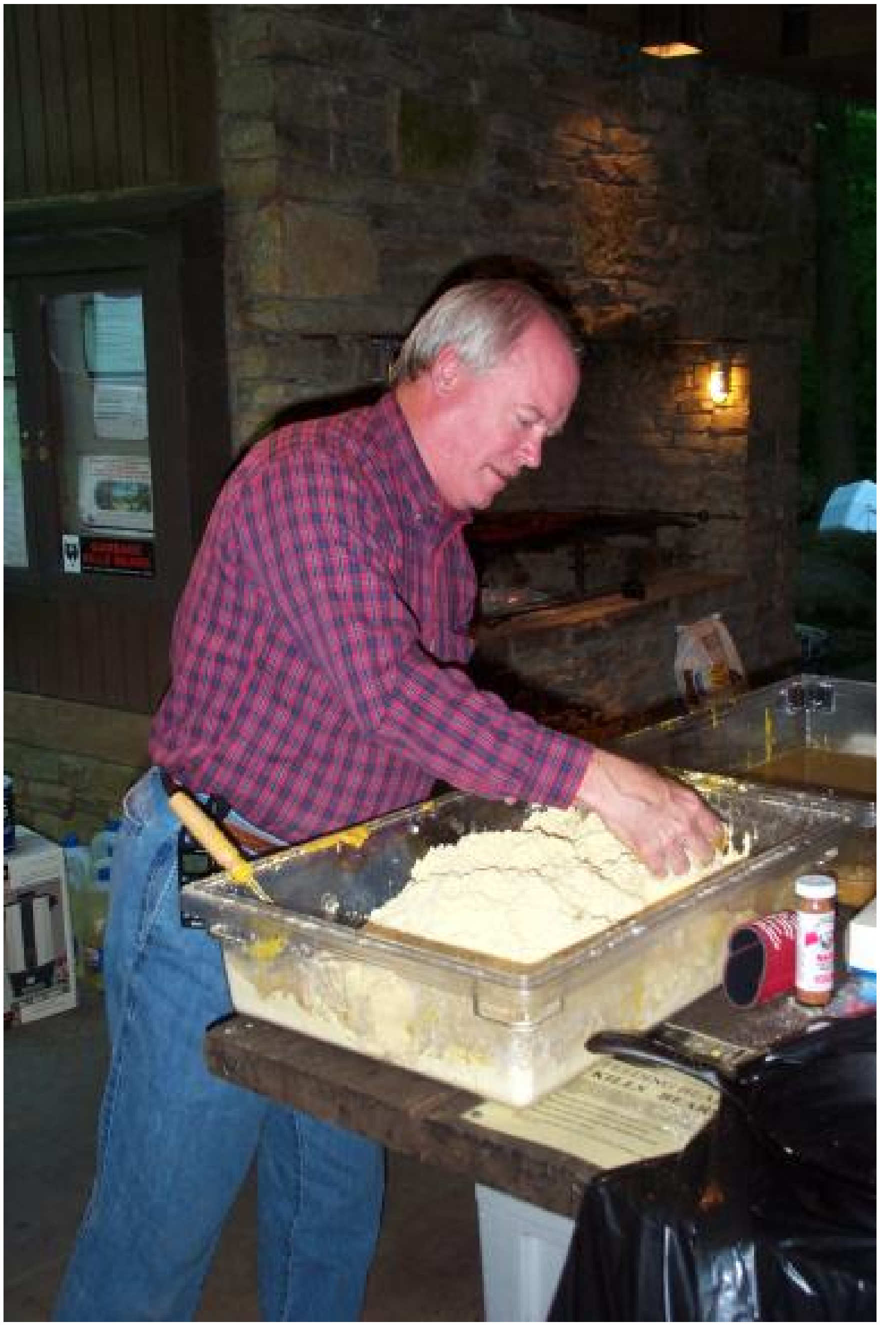


Carl Barnes in the coveted position of stirrer of the corn maque choux. Looks smug, doesn't he?

The ritualistic passing of the sausage. Debate continues on whether this practice pacifies the crowd while the food is cooking, or just revs everyone up. Below, Rich Dafler slices and dices.

Oh sure, they look peaceful enough on the outside. But don't let appearances fool you. It ain't hamburgers and fries they want.

Paul Friedman preparing to call in an air strike if the food isn't ready on time.



Skippy batters the fish. Okay, so this is one meal that isn't untouched by human hands.

Brad and Otis - new for Team Beer this year: uniforms! Expensive, too. The pink elephants are not real.

Bob Dobbins wanting to know why his take home bag has "Bob Mr. Punctual Dobbins" written on it.

The menu was crawfish etouffe with pan fried catfish over rice, corn maque choux, fried pickled okra, and cornbread. There was fried eggplant for the vegetarians, and the kosher folks enjoyed blackened red snapper

Okay, so everyone was too busy wolfing down the meal to take any actual pictures of the main course. 'Nuff said.

Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III: OOP - It's Not POO Spelled Backwards

by Mark Riffey

Published 2002-05-24

Russ Eggen had the difficult task of filling the gap between lunch and the much-anticipated Cajun cookout. His plan: to fill our afternoon with the basics of OOP so that any non-OOPer would be at least cognizant of classes, methods, properties and interfaces before heading out to the mountains for some etouffee.

Russ started by discussing his experiences in teaching OOP and ABC, beginning with his the first TopSpeed ABC class using Clarion 4 beta 3. He explained that the course he was about to teach us was built using his teaching experiences in the years since that beta 3 class.

He questioned the Clarion community's use of OOP, discussed the oft-encountered resistance to doing so, and compared the Clarion templates to the ABC templates. He referred to the large investment in time that many of us have in learning the ins and outs of the Clarion template chain, and he combated that, with a brief discussion in the benefits of using OOP.

It's not POO spelled backwards

A few folks (no, certainly not me!) were intent on asking Russ a few incredibly simple questions just to razz him a little, such as "What's the 2nd O in OOP stand for?" As usual, Russ had thought of all that and ruined our fun. He started at square one, including the definition of the term associated with each letter of the OOP acronym. For example, we arrived at a definition of OOP that stated: OOP means the "writing of code directed or aimed at objects."

Buzzword city

In order to avoid inundating the group with buzzwords, Russ spoke in general non-computing terms when making definitions. Some examples:

- Encapsulation - "To enclose something, as in a capsule"
- Class - a class *is* encapsulation, i.e. Dog and Cat could be classes, of course Animal could be a class as well, but it would likely have Dog and Cat Interfaces. Let's just not go there.
- Property - for a person, a property might be "Height", "Age" or "Skill Set".
- Instantiation of a class - the creation of an object. Instantiation of the house results from its blueprints. Russ described three ways to instantiate a class, then described when to use each instantiation method.
- Dot syntax - how to tell which object owns which property or method, since dot syntax requires use of the object and class name in the statement. For example: Classname.Objectname.

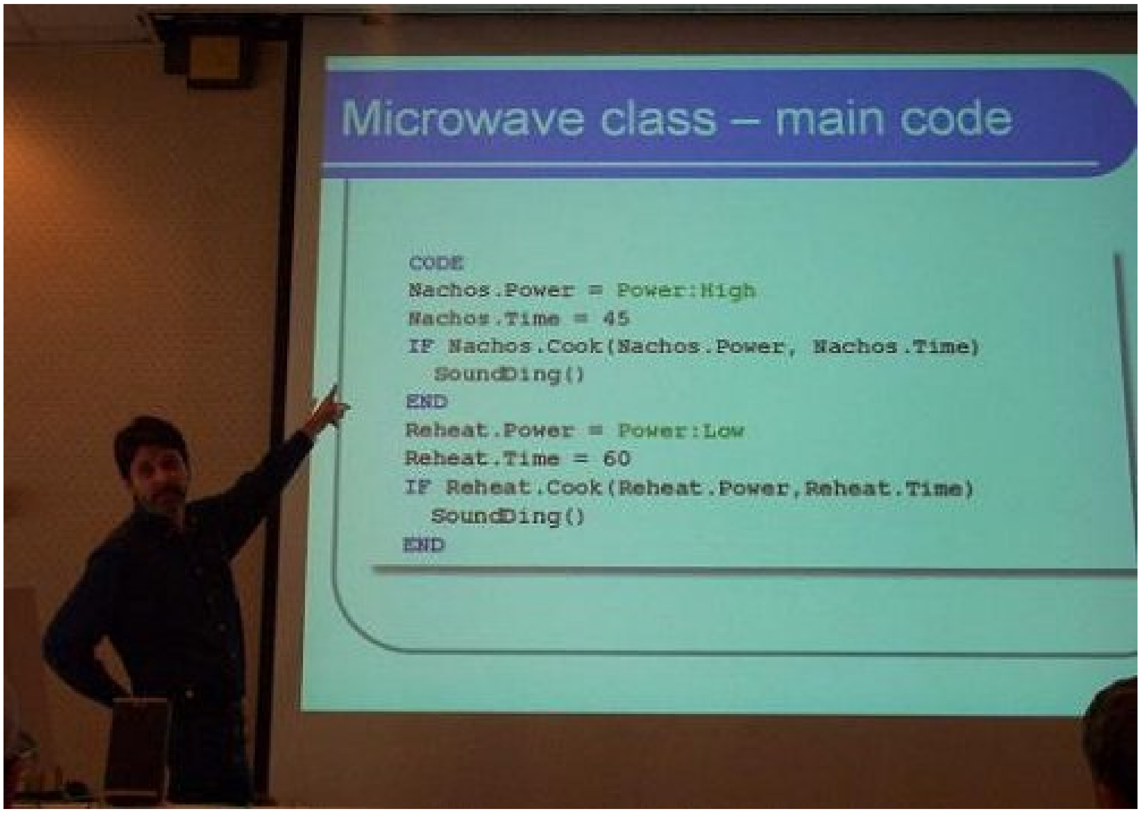
Self.LoveThy

A common question is when to use Self, or for that matter, what in the world SELF is. Russ made it simple. Self is defined as whatever the *current* object is. When coding *methods* (not the main body of the class), the object name is `self`. You do this so that the code is always working with the proper instantiation of the class.

Coding for the microwave

As often happens in newsgroup threads, the discussion ended up at food again. Russ illustrated the benefits of OOP by using his microwave . He made the following statements, and then defined a class for his microwave so that he could handle these tasks.

Russ says:
I like nachos.
I microwave nachos.
I heat my coffee
Cook chili
Melt cheese
I only own 1 microwave



Microwave class – main code

```
CODE
Nachos.Power = Power:High
Nachos.Time = 45
IF Nachos.Cook(Nachos.Power, Nachos.Time)
  SoundDing()
END
Reheat.Power = Power:Low
Reheat.Time = 60
IF Reheat.Cook(Reheat.Power,Reheat.Time)
  SoundDing()
END
```

Do you need multiple methods for cooking? No, you simply build one cook method and create multiple instantiations of the microwave class. You might instantiate microwave classes called Melt, Reheat, Cook, Warm, etc.

For example:
Melt class(microwave)
Reheat class(microwave)

And so on…

After illustrating the use of his microwave class and demonstrating that he could use it over and over again via his melt, reheat, warm (etc.) objects, the benefits of OOP were obvious to all: Code re-use, the existence of a single point of maintenance, and smaller code, even when compared to functions (a function is a procedure with passed parameters).

Construction zone

A commonly misunderstood aspect of OOP is Constructors and Destructors. Russ explained that they are simply special methods in a class. They are automatically called when a class is instantiated (built) or destroyed. They are the startup and shutdown "autoexec" of a class, so to speak. An ETC attendee asked about the difference between `Init` and `Construct`. Russ indicated that the key to knowing when to using `Construct`/`Destruct` instead of

`Init`/`Kill` is that `Init`/`Kill` can be passed a parameter and they can return a value.

Private - keep out!

Russ discussed that a class often encapsulates (hides) data. Clarion uses the Private attribute to perform this "hiding". No code outside of the class can use private properties and methods. He indicated that the proper time to use Private is the times when the class's method or the property is no one else's business outside of that class.

Who's your daddy?

The whole point of inheritance and derivation is to let the programmer enhance or otherwise change the parent class's behavior without changing the class itself. After a short discussion of multiple inheritance (a derived class that has multiple parent classes), Russ indicated that this is not supported in Clarion. Russ indicated that a lack of multiple inheritance saves you work since you don't have to figure out who the parent of the class is..

In a child class, the data type of a property is a reference to another class. No ambiguity of who's your daddy, but all the benefits of multiple inheritance.

For example:

```
Applepie    class,type
Apples        string(20)
Crust         string(20)
            End

Icecream    class,type
Flavor        string(20)
Scoop         procedure
            end

Alamode     class(applepie)
Ontheside     &IceCream
Serve         procedure
            End
```

The `Alamode` class uses composition, i.e. its parent class is the `Applepie` class but it also uses the `Icecream` class.

Protect your privates

Russ went into an extended discussion of moderate encapsulation. In Clarion, properties and methods with no attributes are public. If you wish to restrict access to these items to just the class, then use private. Sometimes you don't want public or private, which means it is time to use Protected. You can access protected properties and methods in your class and in derived classes, but not from any other classes.

Not a parrot with a drug problem

At this point, Russ discussed the definition of a few more terms, including that "polymorphism is not a parrot with a drug problem."

- Polymorphism - basically means "many forms." Clarion has supported polymorphism for years. A close relative of polymorphism is "function overloading", such as Open(window), Open(file). Not really OOP-style polymorphism, but it fits the definition.

- Virtual methods - according to Russ, this is the one thing that drives procedural coders nuts. The Clarion definition - a method whose prototype is present in a parent class and a derived class and both have the ",Virtual" prototype attribute.

The magic of virtual methods is that they call *down* or *forward* to a child method. It is reverse inheritance. In terms of the ABC templates, *every* embed is a virtual method of its respective ABC object. This means you can change the parent's behavior and never touch the parent class's code. The parent calls the child's method instead of its own code. An empty method in the parent class is *meant* to be overridden by a virtual method.

Virtual apple pie

You'd think this was a newsgroup thread, as we spent much of the discussion of OOP in the context of food. An example of virtual methods: in this case `CreateCrust` and `MakeFilling` are virtual methods.

```
Applepie      class,type
Preparepie      procedure
Createcrust     procedure,virtual
Makefilling     procedure,virtual
              End

Dutch         class(applepie)
Createcrust     procedure,virtual
Makefilling     procedure,virtual
              End

  Code
  Dutch.preparepie

Applepie.preparepie proc
  Code
  Self.createcrust()
  Self.makefilling()
```

This allows the dutch apple pie to have special capabilities when preparing the filling (adding special ingredients) and creating the crust (adding crumbles).

I'm all bound up

Often times in a discussion of COM/OLE, a discussion of early or late binding comes up. Russ's explanation for these:

- Early or static binding is where all procedure calls are resolved at compile time
- Late or dynamic binding is where all procedure calls are resolved at runtime.
- Late binding can take a performance hit in other languages. The price is huge in Visual Basic, and in some cases, C++. In Clarion, it's one extra memory table lookup and the overhead is done, i.e. it's quite small.

Interfaces

As Russ stated quite simply, an `Interface` is a collection of methods implemented by a class. Note that there is no reference to the word "data" there. Russ asked at this point how many people considered themselves `Interface` gurus. Only 2 people (Jim Kane and Nik Johnson) admitted to feeling that familiar with interfaces. Russ didn't include himself in that group, but he obviously was comfortable with them at the very least.

- What are `Interfaces` - they are definition of a behavior.
- What benefits do they bring to a class - you can clone a class and replace all of its methods. You are working with the same object yet all the methods have been overridden.

By now, most of us were thinking - so, uh, why do we want to do that? COM is the most often used reason, but COM was intended for Thursday so we didn't go there.

Instead, Russ asked us to think of a printer and asked the question "How can you make a printer have fax and scanner behaviors….*without* changing the code in the printer class?"

The answer is….interfaces.

Here is a small example, which resulted in about two hours of discussion.

```
Ifax           interface
Transmitdoc      proc
Getdoc           proc
               end

Printer        class,type
Docname          cstring(256)
Printdoc         proc
               End

Extendprinter class(printer),implements(ifax),type
Printdoc         procedure
               End
```

At this point, Nik Johnson commented that one should not get tied into "printer" thinking, so we switched the discussion to dishwashers that fax.

```
Extenddishwasher class(dishwasher),implements(ifax),type
WashAndFax         procedure
DryAndFax          procedure
                 End

Extendprinter.ifax.transmitdoc procedure
   Code
   ! code here
```

Why use interfaces? They can be used to structure an entire set of behaviors for a class before the class is written. If you are working in a team, then you can hand out the predefined interfaces for a non-existent class, and everyone's code compiles regardless of how many of those methods have actually been written. "You (the class) will behave like…..". At this point, the interface is described so all this design can be done up front.

An example that might be easier to relate to: in a 24 pin serial cable, all 24 pins may or may not carry a signal, but the socket and the cable still have 24 pins no matter what. A 24 pin serial plug *is an interface*.For example: hardware item A may use pins 2 and 7 (methods), while hardware item B may use pins 8 and 22 (methods). Either way, both devices (objects) support all 24 pins (method definitions), but only use the ones they specifically need.

Due to some technical difficulties with a beverage container access device, Russ continued his presentation until a little after 6 p.m., using the time to cover a pile of examples. At this point, we launched into a discussion of interfaces using Steve Bottomley's applepie interface code, which served as a good deal of the content for this part of the session.

At this point, everyone suddenly felt brilliant (well, sort of) and little light bulbs were popping on all over the room as everyone "got it." But five hours of talking about pie had everyone hungry so we were off to Andrew and Sabrina's cooking and Team Beer's serving….

Mark Riffey has worked in the software industry, primarily in development and technical support for two internationally known enterprise software vendors, the world's premier information systems services company, a Fortune 100 manufacturer, and now Granite Bear Development. His business philosophy is simple: Be fair to your customers and yourself, surround yourself with brilliant people, work hard, be a good listener and have a little fun. Mark and his wife Jacki have two boys, Alex and Jonathan. Mark's other interests include Boy/Cub Scouting, backpacking/hiking and almost anything else outdoors, classic blues guitar, golf and photography.

Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III: A Cajun's View

by Andrew Guidroz II

Published 2002-05-27

We really had a blast at ETC again this year. We arrived Monday night about 9:30 PM and, miraculously, Tommy Hebenstreit popped out of the elevator just in time to help carry some Cajun refreshments and snacks upstairs. The dialog was hard to follow but it sounded a lot like two cartoon characters from MTV. "I am the Great …." "Donde esta tu hallpass?" It wasn't long before we had all the food for the Cajun cookout locked away in the big coolers at the Edgewater Hotel.

Then, up to Lee White's room. Brad Noble, met me with a cool sippy in a glass that would look small only in his hands. The newly appointed second member of the drinks and refreshment group, Otis, also joined us.

It was a nice, early evening for me, in ETC terms, and I was in bed easily by 2 a.m.

I was in the lobby for 6 a.m. the next morning and there already were a few attendees down there. By 8 a.m., there were no less than 30 people running around in the main conference room. That might seem to be a small crowd, but this was two hours before registration was scheduled to open and 10 hours before the opening reception. It was great to see faces from two years before and four years before.

The interesting thing to me about this conference is all the new faces and all the people who say they either never attended a conference before or haven't attended one in over five years. This seems to signal an interest in the new things that SoftVelocity has in the works.

Russ and I made the obligatory trek to the Smokey Mountains Brewery and Grill with 18 of our closest friends and family and, in a few minutes, the place was teeming with Clarionauts. It looked like no less than 40 of us were in that place. The food was good and most of us ordered the beer sampler. The band started up and the party really picked up. Then, one of the most incredible one-man plays began.

Master thespian Mark Riffey took the stage with only two props: a pair of pint sized glasses. He placed them on each side of his neck and, before our startled eyes, began a shuffling walk. He wasn't just acting like Frankenstein, he *was* Frankenstein. Then, without a change of costume or curtain cover, he put the glasses on his head, pawed the ground with his right foot and snorted. He was the ultimate symbol of virility and power: a bull. Then calmness came over him and he placed the glasses on his chest. Here we all leaned forward, aware that this was to be his first speaking part of the night. In a powerful falsetto, he sang,

"Like a virgin
Touched for the very first time
Like a virgin
When your heart beats
Next to mine"

We cheered. We laughed. We cried. I slapped Russ for not having the video camera on.

The tasty drink and chocolates from Norway that Richard Rogers provided were definitely the palette highlight of the trip. Michael Lawson spent some time visiting with us late one night in my room. The conversation was so intense that no one noticed Carl integrating standard XML objects into all my apps.

There were educational sessions somewhere in all of this. Russ had some great stuff on COM and OOP. It was nice to take his picture after he videoed every one of us (I'll trade food for original tapes). Andy Ireland proved that someone can be very articulate, very funny, and a good educator even with all of those gears sticking out of his head.

Bob Zaunere gave us a good feel of where the product is going and I think everyone liked what they heard. He told me that he is considering doing regional meetings to introduce new technology to Clarion users and I think that is a great idea. Bob Foreman did a great job too.

Carl Barnes was his incredibly helpful self at the conference. He was still in his version 2 mode: that is, the weight lifting Carl Barnes rather than the old fat and dumpy version 1 Carl Barnes. I guess root beer and pretzels keep a man fit. He even offered to let me speak to his wife Debra on the phone. It wasn't until the last DevCon that any of us knew Carl was married. With complications that arose at the Cajun cookout, much of the stirring was left to Carl and, as usual, he came through. I think he only did 40 impressions of Fat Bastard. "I'm bigger than you, I'm higher on the food chain. Get in my belly!"

Steve Parker and Shawn Mason did great jobs too. Who would have thought that Shawn was really Spiderman? Or did I misunderstand something? I mean, a guy wears tights and wrestles? I saw the movie. Steve provided me with some lovely pastrami and mustard that has cemented my need to visit Chicago. I even begged Carl to take me back there with him in his plane so I could have a meal there before I returned to Louisiana. Carl didn't have Steve's address for Sabrina to forward the divorce papers, so I decided instead to return to Louisiana with her. One good turn of pastrami deserves another and Steve got to try some authentic blackened red snapper. Ah, the penalties of eating kosher.

The Cajun cookout seemed to go pretty well. I had lots of help from Rick Dafler who had to take over the sausage grilling. This man has prepared a can of soup that can sit on a shelf for six months and has prepared it in batches for tens of thousands. *That* takes more guts than cooking for the 175 folks scheduled at the conference. Mike Hanson came to me that morning and offered us the use of his car, and his wife Jill Happenstance's help in preparing the meal. If that ain't Super, I don't know what is. Jill was a pleasure to work with and convinced us to spend some time at the aquarium later in the week. I don't remember seeing more fascinating stuff. And I don't remember a craving for Red Lobster coming on that strong either.

At the cookout, there were no bears. Team Beer worked furiously and even served up many cases of soft drinks. We were late starting but ready to serve at 7 p.m. Oddly, after looking around, I realized that I wasn't going to interrupt the crowd. We waited a bit to serve because folks were just too busy visiting and laughing to eat. *That* is when you know things are going well.

We got a ton of help at the cookout. Skip Williams was Sabrina's first assistant in the battering and frying of the fish. His child bride Sharon was generous offering him because: 1) they still behave like honeymooners and letting him away from her side for that long must have been tough; and 2) she doesn't like the taste of catfish, and Skip rubbed his hand on nearly 250 fillets. Our son Joseph was everywhere helping to coordinate a bunch of non-Cajuns in the methods of preparing a Cajun feast. Maybe next time I'll sit back and let him cook the whole meal. Being able to cook a meal at a big event is when you know you are all grown up in our culture.

Bobby Bass helped us serve along with Bob Dobbins. Lee White cooked an excellent hillbilly cornbread that I made sure to enjoy. Bob Campbell once again reprised the roll of cleanup man. It is no small feat to clean up after a crew this size and Bob always is the first to help. There is no one the cook likes more than the cleanup crew.

Leroy "That's My Leroy" Schultz chased around with us. Michael Lawson and Chris Kloss demonstrated that they could last with the more experienced ETC partiers. Michael Brooks was even seen enjoying a Cajun beverage, although he didn't stay up late with us. Next time, we'll find a local band that knows a few songs off the "Sacred Fire" CD.

Carl Barnes and Tommy Tequila drug us screaming and moaning all the way to Hillbilly Golf and we actually got to play. Bruce Johnson and Gary James proved that South Africans are *really* too serious about golf at any level. Bernard Grosperrin nailed a nice hole in one. It was something new for us to see this motorcycle roar up, a man in full leathers jump off, rip off his helmet and say "Bonjour!"

Drew Bourrut showed courage by being the only vegetarian at the cookout. I think we succeeded in providing a Cajun menu that had no pork. Pratik Patel, the Purple Benze man himself, showed me a new third party product that knocked my socks off. That's just a teaser for y'all … expect to see it in the next few weeks.

An early breakfast one morning with Nik Johnson and Stamos Fafalios proved to me just how technology dependent we Clarion programmers are. No one carried a GPS, and we were lost within moments. Thanks for the apron, Stamos. One day I hope to try that Greek goat recipe you and I talked of before.

Lee G. White had a fantastic conference organized again. People forget that the origins of this conference all go back to a certain Cajun and Hillbilly who got to be friends on the phone and decided they wanted to see each other in person. Lee cooked a meal for me and I cooked one for him that time and it helped cement a great friendship and love between our families. Later, I vowed to never cook for another conference and Lee coaxed me out there to cook again. Always remember that when I'm cooking there at the conference, the serving may be many, but it is all still me cooking a plate for Lee.

My wife Sabrina, son Joseph, and I had a great and wonderful time. Everyone was just way too kind to us again, and we really appreciated the opportunity to visit and share a little Cajun culture with y'all. I hope I remembered everyone who helped but I definitely know I didn't mention everyone who I talked a bit with and shared some time with. We'll see y'all there again in two years … or maybe just a little bit sooner.

Andrew Guidroz II, when he isn't traveling around the countryside watching his 2001 SEC Champion LSU Fighting Tigers, writes software for all facets of the insurance industry. His famous Cajun cookouts have become a central feature of Clarion conferences throughout the U.S. Andrew's Cajun website is www.coonass.com.

Reader Comments

Add a comment

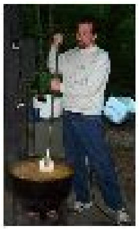
Clarion Magazine

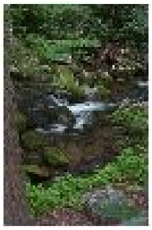
Home COL Archives

Topics > DevCons > ETC III

ETC III: Photos By Mark Riffey

These are thumbnail images of the full-sized pictures. To view a picture, or to add your own comments to a picture, click on the thumbnail. Any comments you add will appear on the Reader Comments page along with a link to the image (unless you specify a private message to the editor).





Reader Comments

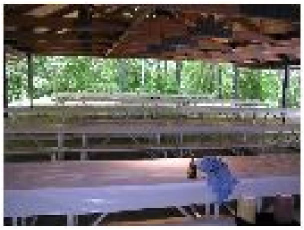
Add a comment

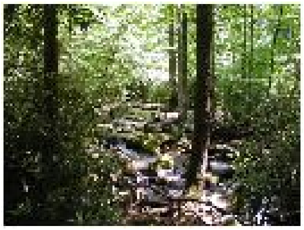
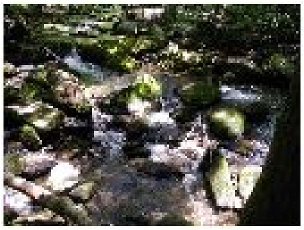
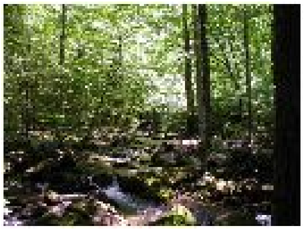
Clarion Magazine

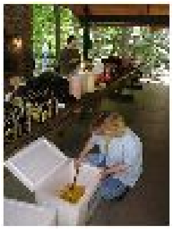
Topics > DevCons > ETC III

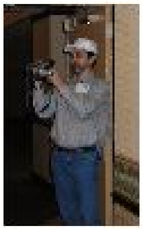
ETC III: Photos By Andrew Guidroz

These are thumbnail images of the full-sized pictures. To view a picture, or to add your own comments to a picture, click on the thumbnail. Any comments you add will appear on the Reader Comments page along with a link to the image (unless you specify a private message to the editor).







Reader Comments

Add a comment

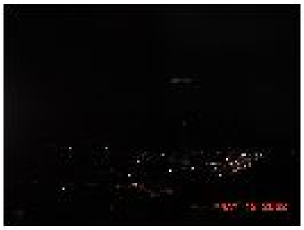
Clarion Magazine

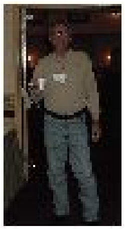
Home COL Archives

Topics > DevCons > ETC III

ETC III: Photos by Brad Noble (1)

These are thumbnail images of the full-sized pictures. To view a picture, or to add your own comments to a picture, click on the thumbnail. Any comments you add will appear on the Reader Comments page along with a link to the image (unless you specify a private message to the editor).





Reader Comments

Add a comment

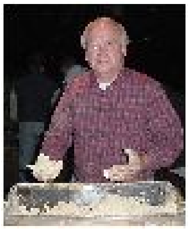
Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III: Photos By Brad Noble (2)

These are thumbnail images of the full-sized pictures. To view a picture, or to add your own comments to a picture, click on the thumbnail. Any comments you add will appear on the Reader Comments page along with a link to the image (unless you specify a private message to the editor).



Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

Thanks! From Lee White & KCUG

Published 2002-05-28

The Knoxville Clarion Users Group wishes to thank all those who attended the third East Tennessee Clarion Conference and Gathering. In addition to the attendees we thank all the presenters, representatives of SoftVelocity, and third party sponsors without whom this event would not have been possible.

It was truly a pleasure to once again gather with old friends and new.

On a personal note, I want to publicly thank two others who provided assistance prior to and during the conference. Brad Noble and Norm "Otis" Owings, team beer!, helped in numerous ways that provided me with the time to enjoy this conference. To them both I owe a debt of gratitude. Thanks guys.

Here's to a very prosperous two years in anticipation of the fourth etc gathering.

Lee White
Knoxville Clarion Users Group

Reader Comments

Add a comment

He's thanking US ??? Hey Lee, stop being so modest and...
The pleasure was all mine. I believe what was said at the...
Thanks Lee for another great conference! I can hardly...
Most of you, Lee not included, have no idea how much I...
No, Lee...the pleasure was ours. We really enjoyed our...

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III: ClarioNET and More - Bob Foreman

by Nik Johnson

Published 2002-05-30

When Bob Zaunere finished his ETC III keynote address, two hours longer than scheduled and light years beyond my expectations with respect to content, there seemed to be little left for Bob Foreman to add. But while the ASP templates have been announced before, and ClarioNET has been available for some time, Bob's presentation added substance that only an actual demonstration can provide.

ASP and ClarioNET nicely bracket Clarion's growing set of options for server-side processing. (For a concise description of these options, see Tom Hebenstreit's article SoftVelocity's Upcoming Product Technology (Part 2)).

ASP

Bob walked us through the creation of a simple app and its adaptation to use ASP. On the surface, at least, the steps are pretty straightforward:

1. Create a directory for the app. Structures should be those of a SQL DBMS.
2. Use the IIS Virtual Directory Creation Wizard to create virtual directory for testing.
3. Add relations, business rules, additional indices, control formats, and descriptions for tables and fields. This dictionary work is particularly important. First, validity checks in the dictionary allow the templates to perform client-side validation, using a library of JavaScript functions. Also, column properties are picked up from the dictionary.
4. Optionally, adjust field order within each table and set "do not populate" flags as appropriate. If you're using the application wizard, this avoids the necessity of later tweaking. The ASP templates read list boxes and windows to determine which columns should be populated and in what order.
5. Create the application, either by hand or by using the application wizard.

6. Add the ASP Global Template and specify the DSN, the HTML design-time template, and style sheet options.
7. Add ASP Browse or ASP Form templates to the procedures for which you want to generate ASP page equivalents.
8. Use the design-time interface to customize fields if necessary.
9. Generate the app, which also generates the ASP pages.
10. Test.

The templates are designed to permit query-form and form-only patterns as well as the traditional Clarion browse-form scheme. So you should be able to control access as your situation demands.

Bob said that the templates are well documented, with many examples, so you should be able to develop useful web-based ASP applications without having to become expert in HTML markup and scripting languages.

In the course of his talk, Bob produced and demonstrated a working example based on the Northwind database. I came away with the feeling that I could do it too.

ClarioNET

The ETC schedule listed Bob Foreman's presentation of ClarioNET at 10AM. Given Bob Zaunere's extensive opening and Bob Foreman's ASP demonstration, it was not a big surprise to find ClarioNET appearing stage center at 12:52 p.m., dangerously close to scheduled return flights for the two Bobs and lunch for the rest of us.

It should be noted that several comments were made concerning the tendency of SoftVelocity to hire people named Bob. We were assured that this is accidental. I also learned from a reliable source that there are no plans at present to develop a proprietary web-based tool named Bob-O-Link.

ClarioNET approaches the web quite differently than do the ASP templates, although both start with a conventional Clarion dictionary and application. ASP involves server-side code which generates HTML pages, which in turn are read by a browser at the client side. ClarioNET uses a thin client program to mirror the activity of a Clarion program running on the server. The result is an application on the client machine which looks and feels very much like a Clarion desktop application.

While demonstrating a ClarioNET application, Bob pointed out some limitations. The following are not supported at the present time:

- Edit-in-place.
- QBE.

- Pause/Cancel buttons on progress windows.
- OLE/OCX controls.
- Some third party templates. (CPCS, RPM, and Fomin reporting templates are explicitly supported.)

ClarioNET also explicitly supports the ABC and Clarion template chains (Menu, Browse, Form, Report, Process, Source), and additional support for third-party templates is added regularly.

There is also a “Push” window which can be used for progress displays and the like with Open/Update/Close functionality.

Nik Johnson stumbled into the programming racket in 1959 when his boss at Grumman Aircraft insisted on his attending a Fortran class. Since 1986 he has been using Clarion to help clients solve information handling problems.

Reader Comments

Add a comment

Clarion Magazine

Home COL Archives

Topics > DevCons > ETC III

ETC III: Programming in Clarion's ABC - Bruce Johnson

by Brice Schagane

Published 2002-05-30

Once again, Bruce Johnson returned to the ETC conference for his presentation of Programming in Clarion's ABC.

Bruce has expanded his previous presentation at ETC 2000 to cover significantly more information. For example, attendees of this session received a 137 page book, written by Bruce Johnson, that details all the information discussed. The book for this year's session is approximately three times bigger than before.

Bruce opened with a discussion of some basic ABC terminology and outlined some goals of using ABC, the primary goal being: writing reusable, bug free code.

The information presented this year was simply overwhelming. The topics that Bruce covered included: Windows, Forms, Browses, Views, Reports, Processes, Files, Relations, Errors, Popups, INI files, and Edit In Place. Each topical discussion focused on the ABC Classes involved and how they relate to each other.

Bruce explained the relationships between "Generic" and "Specific" code and helped grasp the concept of "Virtual Methods", or "Pancakes" as he calls them. If your confused by this, don't be alarmed, I was too. Bruce was basically implying that the term "Virtual" is too vague, and should have been called something more meaningful.

Bruce also discussed Interfaces. He explained what they are and how to use them. In each of his discussions, Bruce pointed out major embed points and when we should use them. Along the way, Heprovided many useful tips and tricks. He also presented some little know facts, including:

- How the Update method of the FileManager Class can cause a performance hit
- How deferring the opening of files until accessed can provide a massive speed improvement
- Why embedding code into the Insert() method of the FileManager Class should also be embedded into the TryInsert method as well
- How to view error log history and/or save it to disk
- How to get a list box to behave like a spreadsheet
- What the C55util function library is and how to use it

As an added bonus, Bruce will be providing several resources to the participants, via the web. This will include some examples, as well as custom `zError` and `zINIClass` classes. These classes overcome some of the limitations of their respective ABC classes.

All in all, too much information was discussed than can be included in this report. At times, even Bruce skimped on the details, but would direct us to read the book, because the information is discussed there.

There was an additional cost of $50 to attend this session. Even if you were unable to attend, it is my opinion that the book alone is worth the money spent. The book is laid out quite nicely. The first section is intended to be an introductory tutorial, while sections two and three cover more advanced topics. Compared to the cost of many computer books these days, the cost to attend this session was a bargain. If you have the opportunity in 2004, you should definitely plan on attending this session.

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

Reader Comments

Add a comment

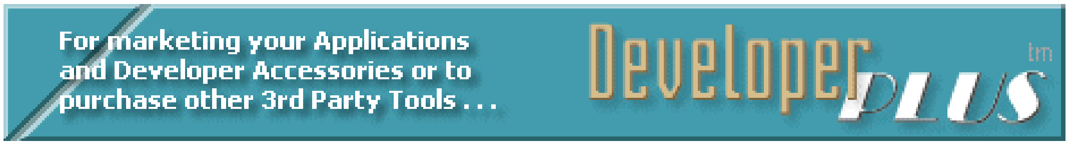
Sounds as if Bruce's session is one not to be missed, I...
Please add me to any Book List. To Quote George McFly,...
The contents of Bruce's presentation and book are exactly...
I would very much like to buy a copy of the book...
Would be very interested in purchasing a copy of the book...
How can I get a copy of Bruce Johnson's book?
I too would be interested in purchasing this book.
Does Bruce have any plans on selling his book? I would be...

I also would like to know if the book will be for sale?

I too would like to get a copy of Bruce's book ... in any...

My main reason for wanting to go to ETC was COM and...

Clarion Magazine



Home COL Archives

Topics > DevCons > ETC III

ETC III: Reports: Paying the Price of Flexibility - Steve Parker

by Brice Schagane

Published 2002-05-30

Steve Parker presented the sixth and final session of ETC-III conference. Steve has authored more than one hundred Clarion articles over the years. Of these, twelve have dealt with printing and reporting in Clarion. Steve opened the session with his typical humor. When it comes to reports, there are two things to remember (everything else is irrelevant):

1. RPM with ReFlex
2. CPCS

While Steve doesn't use these templates himself, if they work for you, why change? This falls back on the old saying, "If ain't broke, don't fix it."

Reporting may be considered by some as a nasty topic, but as Steve pointed out, it is highly relevant. Reports are the single most important part of any application. This is specifically due to management: since the primary audience of reports is managers (and they sign the checks), it's important to keep them happy. Saving time saves money, and this pleases management. The better your reporting capabilities, the greater the savings.

After a brief history lesson on reporting in DOS, Steve pointed out the many problems that Windows introduced to the reporting process. In Windows' attempt to simplify the reporting process, it has removed the ability to control several important aspects of the reporting process. One of these is page overflow.

Steve made many comparisons between reporting under DOS and Windows. With the functionality that is lost under Windows, Steve pointed out we need to find something similar. The closest thing here is the View engine. Views provide the ability to sort by anything and break reports at any time. Everything is exposed to the programmer inside the View engine.

This is where the flexibility lies.

Steve also stated there are many deficiencies in the report template and emphasized that you don't use it. If the standard capabilities are all you need, then the price of flexibility is nothing. However, if your needs are more exotic, the price is three-fold:
1) Understanding View specific ABC methods
2) Understand branching logic and the PRINT( ) statement
3) Mastering a few embeds and a couple of template options

Someone once said, "When it comes to programming, everything of importance happens inside a loop." Likewise, reports utilize a loop for processing. Steve broke the reporting process into four simple steps:

1. Create and Open the View
2. Open the Report
3. LOOP
4. Print end of report stuff

After creating and opening the view, you should set any filters and sort orders. In this section, Steve demonstrated many useful methods for generating some highly complex reports.

The next step is to open the report. Here, Clarion checks to see if there are any qualifying records. If so, the report is opened and processing continues. If not, the TakeNoRecords method is fired. Inside this method, you could print a "No records available" message.

Next is the heart of it all. Inside the loop, you can perform additional filtering, test for breaks, and process any totaling. Steve pointed out several useful embed points and revealed several tricks for handling breaks. Jim Mumford also provided a trick for printing a "Continued…" header message when page overflow occurs. He also hinted at the possibility of writing such an article in the future.

Once the loop is complete, you can print your end of report stuff. This may include totals, footers, or any other comment sections.

Steve's greatest companion is the Pause Control template. This template provides you with the means to collect vital information from the user for the creation of a truly customized report.

In summary, Steve stated that the report template does not do as much as you would expect. However, the View engine does, and for this reason, you should use it!

Steve's presentation was highly informative for both the novice and experienced developer. If you missed it, I'm sorry. The information obtained from this and past ETC conferences has been well worth the money. If you have not yet attended an ETC conference, you should consider doing so in the future. The next one is scheduled for 2004.

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

Reader Comments

Add a comment

Thanks for the nice review, Brice. I do have to mention...