Reborn Free

CLARION *online*

published by **CoveComm Inc.**

Clarion MAGAZINE

TopSpeed

Clarion 5 *by TopSpeed*

FREE Microsoft Internet Explorer

etc2000 EVENT SPONSOR

May 2000 Index

Clarion: Back Where It Belongs
Clarion Magazine publisher Dave Harms comments on the recent acquisition of the Clarion product line by a group of investors headed by Robert Zaunere, previously TopSpeed's VP of Product Development.
(May 4, 2000)

The Cranky Programmer: Survey This!
Cranky tried to take the weekend off, seeing as how ClarionMag is on one of its rare holiday weeks. But then he made the mistake of taking TopSpeed's development strategy survey…
(May 2, 2000)

Template Viewing Utility
Jo Baron's freeware template viewing utility makes it easy to read template code. Requires Linedraw.TTF.
(May 2, 2000)

Clarion COM: As Easy As Visual Basic COM
Can COM be as easy in Clarion as it is in Visual Basic? Impossible, you say? Read on.
(May 9, 2000)

Clarion Challenge: COM Examples
In this week's issue Jim Kane explains how Clarion COM can be as easy as VBScript COM. Show us what you can do with Clarion COM scripting.
(May 9, 2000)

Cloning List Boxes - Source Update
The Multibox source code has been updated with several bug fixes.
(May 9, 2000)

Interview: SoftVelocity's Bob Zaunere
On May 2, 2000 Frank Watts, TopSpeed's President and CEO, announced the formation of two separate, independent companies to continue TopSpeed's business directions. One of these is SoftVelocity Inc., which takes over TopSpeed's Clarion line of rapid application development tools. SoftVelocity's President and CEO Robert Zaunere spoke with Dave Harms, Clarion Magazine's editor and publisher.
(May 16, 2000)

The Novice's Corner: File Access Basics
Almost all Clarion applications work with data files of one description or another. Most use databases with numerous different tables and files, and perhaps even more than one database or file system. And all of this is ultimately possible because of Clarion's file access grammar.

(May 16,2000)

[Product Review: LSZip 2.5 From Linder Software](#)

Most of us use a standalone shareware or commercial application to do all kinds of the zip file-related work. There are times, though, when it would be useful to be able to manipulate zip files directly from within a Clarion program. Enter LSZip from Linder Software.
(May 16,2000)

[Next Week: ETC Coverage!](#)

Next week is the East Tennessee Clarion Conference in Gatlinburg, and Clarion Magazine will be there, bringing you regular updates on the conference proceedings.
(May 16,2000)

[ETC Starts Off In Style](#)

The East Tennessee Clarion Conference got rolling Tuesday with an excellent presentation by James Fortune, who soldiered on despite technical difficulties. Following that was an evening reception accompanied by a string quartet and all kinds of amazingly good grub. Bob Zaunere, Dave Harms, and Nik Johnson spoke on Wednesday, amid some pretty radical weather. Pics and MPEGs!
(May 24,2000)

[Bob Zaunere Keynote Address](#)

Well, technically it wasn't a keynote address as Bob wasn't originally scheduled to speak at etc, but he was there and agreed to say a few words and answer questions Wednesday morning.
(May 24,2000)

[TopSpeed SA Posts C5.5 Press Release](#)

Here's a screen shot of the C5.5 press release with the new feature list and a preview of the SoftVelocity logo.
(May 25,2000)

[May 2000 News](#)

Clarion news, notes, and happenings from around the globe.
(May 31,2000)

[Complete Cajun Cookout Coverage](#)

Always an eventful feast, this year's Cajun Cookout got an extra shot of adrenalin when a young black bear made a series of unscheduled appearances. Now with full text and pictures (of everything but the bear).
(May 30,2000)

[Tool Talk: Vendor News from ETC](#)

Clarion Magazine's intrepid reviews editor took some time at ETC to catch up with the major Clarion tool vendors. There is a lot of good stuff on the way.
(May 31,2000)

[ETC Presentation Summaries, Part 1](#)

The first of two installments summarizing the presentations made at the East Tennessee Clarion Conference.
(May 31,2000)

Reborn Free   CLARION *online*   published by CoveComm Inc.

Clarion MAGAZINE

Clarion: Back Where It Belongs

TopSpeed's Press Release

by Dave Harms

And now for the news. A group of investors, headed by Robert Zaunere, previously TopSpeed's VP of Product Development, has purchased the Clarion product line from TopSpeed Corporation. The new company, called SoftVelocity Inc., will continue Clarion product development. TopSpeed's other operations, such as consulting and the Wired On MainStreet venture, will be spun off to Sensium Corporation. Sensium's president is Arthur Barrington, son of Clarion creator Bruce Barrington.

No, that isn't the story I wished I'd written for April Fool's. Actually, come to think of it, it is, because then I'd be a prophet, and not of doom either. This is a good news story, and developers have reacted with delight and congratulations for Zaunere, as evidenced by postings on the TS newsgroups.

Ross Santos, CTO at Legal Files Software Inc. points out Zaunere's long history with Clarion. "[He] has been the champion of the product for as long as I can remember."

That's what Clarion has lacked this past year: a champion. That role used to be Bruce Barrington's, of course. And for many years he guided Clarion through sometimes stormy waters. Barrington has been less visible of late, reportedly involved in other business affairs, and perhaps taking a bit more time off, a benefit richly deserved.

Then at last year's DevCon attendees saw a radically changed TopSpeed Corporation, tightly focused on sales and marketing. And that's not a bad thing – certainly lack of marketing has been an ongoing concern in the Clarion development community. But can marketing succeed if the product doesn't do what developers need it to do? Rumors about Clarion's uncertain future abounded. Where was TopSpeed headed? Would desktop development be sacrificed for Wizatron-driven web development? Who was really driving product development? And in the months that followed, TopSpeed's silence about its future direction raised the FUD*level to an all-time high.

Wizatrons were to be the bridge between Clarion and the web. A brilliant technology, Wizatrons however failed to capture the imagination of developers, and, it would appear, did not solve all the issues in remaking Clarion into an non-technical web development tool. That failure should free significant resources to address current weaknesses in Clarion 5.5, and I hope and expect that some long-desired features will finally make their way into the Clarion toolset.

Zaunere's move to take over the Clarion product line brings fresh hope to Clarion developers. It's a win for the geeks. Let's hope it's also a win for the investors.

Dave Harms

Publisher
Clarion Magazine

*Fear, Uncertainty and Doubt

Reborn Free

CLARION *online*

published by **CoveComm Inc.**

Clarion MAGAZINE

TopSpeed

Clarion5

FREE Microsoft Internet Explorer

etc2000 EVENT SPONSOR

The Cranky Programmer: Survey THIS!

Call me a clam, friends, cuz I am *steamed* today!

After literally years of trying to get TopSpeed to remedy what I (and more than a few others) view as fundamental gaps in Clarion (can you say "OLE/COM"?), TopSpeed recently announced that they wanted some indication from us (the customers) of where they should be going with the product.

Say what? I thought I had been doing that all along – at times loudly, at others gently, but always making *some* kind of noise (and usually about the same items). Unless...hey, wait a minute - I always *thought* that suggestion box looked suspiciously like a circular file!

In any case, leaving aside how they should have been listening in the past, let me move on to their present plan: an online survey created with C5.5 Beta 2. At least they are saying officially that they want our input now, so I was willing to cut them a bit more slack.

Note the use of the past tense, though. Let's just say I have a few issues with the survey.

Issue #1 – Where Is It?

If you want your customer base to fill out an important survey, one that will help direct the future of the company, maybe you should tell them about it.

Well, duh. Except...

The last email I received which mentioned the survey was the Spring developer newsletter, in which Frank Watts (TopSpeed President and CEO) stated:

"Although this survey was expected to be posted on our Web site by the end of March, a

few unforeseen delays pushed the project back a couple of weeks. We look forward to it being ready by the middle of the week of April 17th."

Ok, fine. So it was delayed. Given the importance of the survey, I expected that they would send out another notice once it was up and running. I even kept dropping in at the TopSpeed web site to see if the survey was available. So how did I find out that the survey *was* in fact up? From Clarion Magazine, when I happened to check up on the previous weeks news. (Shameless plug: see, it pays to read the magazine!)

Did they post messages in the newsgroups? I hope so, but can't say for sure myself. I have been too busy lately to wade through the reams of messages posted there. My point here is that I should not be penalized (i.e., miss filling out the survey) simply because a) they don't tell me about it, and b) they don't mention it on their web site.

Come again? As of the time I am writing this column - at least two weeks after the survey became available - there is still *no* link to the survey on the TopSpeed site (at least, that I have been able to find).

Issue #2 – This Is The Best They Can Do?

Given the emphasis on all of the cool new Internet capabilities in Clarion 5.5, I was looking forward to seeing what kind of application TopSpeed could produce. I mean, I figured that if anyone could do something interesting with it, it would be the people who created the tool.

Boy, was *I* ever wrong!

Talk about generic. Talk about sparse. Talk about a total lack of anything remotely helpful when you first view the survey sign-in page. There are no instructions on the page. In fact, the only way I knew that I could fill out the survey once for each Clarion 4 or higher license I own was from the news blurb in the magazine. The survey itself said nothing about that. (As an aside, not letting people with Clarion 2.0 vote seems a funny way to get them to finally upgrade.)

Aesthetically, it would seem that no one bothered to look at the survey after it was on the web, as evidenced by some of the slapdash formatting.

Figure 1. Ummm, try again?

All right, I think, maybe I'm too finicky and was expecting too much. At least the app works, I'm thinking, as I go to the second page of the survey.

Silly me.

Issue #3 – Didn't Anyone Test This Thing?

Before I go on, let me also say that, in my very own personal opinion, the design of the survey itself, well...it sucks.

There, I said it.

How does the survey work? You are shown a list of about forty items, ranging from fundamental (32-bit IDE and real OLE/COM support) to trivial (spin boxes wrapping). You are supposed to assign percentages to various items based on the importance you attach to the item, totaling up to a maximum of 100 percent. This means you either have to spend a whole bunch of time assigning and tweaking teeny percentages on lots of items, or just give up and use all your chips on a few items.

Further, I wanted to know how are these items going to be weighted in tallying the final results. I mean, is 30% assigned to having spin boxes wrap being given equal weight with 30% for creating OLE/COM objects? I would certainly hope not, but there is no way of knowing from the information provided there.

Why in the world didn't they simply let us rank the ten most desired features, 1 through 10?

On the plus side, at least the instructions mention that a running total of your assigned percentages will be displayed in the status bar of the browser to help you keep track of where you are.

Ok, enough – it's their survey and their design. I just have to live with it.

I start to fill out percentages, then take a look at the little total calculator I mentioned above. Hmmm, I seem to have used up "Javascript Error" of my percentages so far.

Looks like no one bothered to test the survey with Netscape.

Rather than try to juggle all sorts of percentages in my head, I decide to cancel and come back using Internet Explorer. I press the Cancel button and see a message thanking me for taking the survey. Hey – but I *didn't* take the survey yet, right? I *canceled*, right?

Fire up Internet Explorer, go back to Clarion Magazine (so that I can get back to TopSpeed to take the survey again), enter the same serial number, and up comes this little gem of a message…

Figure 2. You talking to me? I mean, you talking to ME?

Aaargh!!

It not only didn't cancel my earlier survey when I pressed the cancel button, it is now calling me names – and expecting me to agree with it!

Reluctantly pressing OK, I am told that I have already taken the survey, but thanks again anyway, then… back to the TopSpeed home page, back to Clarion Magazine (to get back to the survey), etc., etc.

Feels like déjà vu all over again, as Yogi the sage said.

Issue #4 – See Issue #1

All right, after all of my travails I finally got through one iteration using the serial number from another of my many Clarion licenses. They thanked me politely, and then sent me back to the TopSpeed home page. Where, of course, I had no way to get back to the survey so that I could do it all again (and again and again).

Sigh. Back to Clarion Magazine to start the whole process anew.

Stick A Fork In Me, I'm Done

So what should you do?

Go fill out the survey, of course!

It may be lame, buggy and needlessly complex, but it is the *only* way to register your opinions on the future feature set and direction of this beast we all make a living from. If you have multiple licenses for Clarion 4 or higher, exercise your right to vote as many times as you can.

And just to make it easy on you, you can click here to do it. Just be sure to use Internet Explorer - and don't try to cancel.

Me? I'm going to go put one of my Clarion boxes in the back yard and throw darts at it for a while (otherwise known as a Cranky stress break)…

Got an opinion? Tell Cranky.

Reborn Free

CLARION *online*

published by
CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5

Microsoft
Internet
Explorer

etc2000
EVENT SPONSOR

Clarion COM:
As Easy As Visual Basic COM

by Jim Kane

I admit at times I have been just a little jealous of the ease of use of COM in Visual Basic or Visual Basic Scripting. I do hate VB Script's slow execution speed and the current lack of OOP, but sometimes I just wish I could use COM that easily. For example, cut and past the code in Figure 1 to a text file and save as Drives.VBS on the desktop. Assuming Windows scripting is installed, when the file is double clicked in Explorer the script runs and displays the total size and free disk space of the C drive. (For more information about scripting have a look at http://msdn.microsoft.com/scripting.)

Figure 1. Save as Drives.VBS and double click.

```
Dim FileSystemObject, DrivesObject, DriveObject, ↵
  TotalSize, FreeSpace
Set FileSystemObject = WScript.createobject↵
  ("Scripting.FileSystemObject")
Set DrivesObject = FileSystemObject.Drives
SEt DriveObject = DrivesObject("C")
TotalSize = DriveObject.TotalSize
wscript.echo Totalsize
freespace = DriveObject.FreeSpace
wscript.echo freespace
Set  DriveObject = Nothing
Set  DrivesObject = Nothing
Set  FileSystemObject=Nothing
```

The code uses Windows scripting to create an instance of the `FileSystemObject`. It then gets the Drives object from the `FileSystemObject` and the C drive object from the drives object. Once it obtains the object that knows all about the C drive, it reads the `TotalSize` and `FreeSpace` properties. Following that, the objects obtained are released or, in VB parlance, set to nothing.

Well, can it be done as easily in Clarion? Of course! In fact, it takes just four lines (assuming, of course, you have that magical oletcl object, which I'll explain in a bit):

```
oletcl.init('Scripting.FileSystemObject')
message(Format(oletcl.getprop('Drives.item("D").TotalSize')|
  ,@n18))
message(Format(oletCl.getprop('Drives.item("D").FreeSpace')|
  ,@N18))
oletcl.kill()
```

Or matching the VB code line for line and adding the usual Clarion code structure:

```
program
_abcdllmode_  equate(0)
_abclinkmode_ equate(1)
 map
   end
Include('oletcl.inc')
oletCl        oleTClType !dim FileSystemObject
Drivesobject cstring(20)!dim DrivesObject
DriveObject  cstring(20)!dim DriveObject
TotalSize    string(18) !dim TotalSize
FreeSpace    string(18) !dim FreeSpace
 code
oletcl.init('Scripting.FileSystemObject',1)
oletcl.getobj('DRIVES', DrivesObject)
oletcl.getobj(DrivesObject & '.Item("C")', DriveObject)
TotalSize = oletcl.callobj(DriveObject,'TotalSize')
message(TotalSize,'TotalSize')
FreeSpace = oletcl.callobj(DriveObject,'FreeSpace')
Message(FreeSpace,'Free Space')
oletCl.releaseobj(DriveObject)
oletCl.releaseobj(DrivesObject)
oletcl.kill()
```

Ok Mr. Wizard, how can that be done with out a window and OLE control? Well actually, the oletcl class starts a new thread with the Clarion `Start()` procedure, and the required window and OLE control are on the new (and hidden) thread. The `oletcl` class takes care of all the inter-thread communications through the creative use of class references. Since all this in a reusable class, it only has to be coded once. The Clarion `Start()` procedure can take up to three strings as parameters. In this case only one is needed.

What I'll do is pass the `oletcl` object, or more accurately the address of the oletcl object, to the procedure with the hidden window and OLE control. I call this procedure with the hidden window and OLE control `OLEHost`. In `OLEHost` I can use the address of the `oletcl` object to make a Clarion reference. It's actually much easier than it sounds and just takes a few lines of code:

```
StrAddress string(30)
StrAddress = Address(OletCl) !or address(SELF)
Start(OLEHost,40000,StrAddress)
```

And in OLEHost:

```
OLEHost procedure(string strAddress)
LongAddr      Long,auto
Loc:OLETCl   &OLETCLType
  code
  !convert address from string to long
  LongAddr = strAddr
  loc:OleTCl &= (longAddr)
```

The last line above is the magic that makes this whole thing work. This is frequently called a cast in other languages. By this means the raw address for the class is converted into a Clarion reference variable that refers to the class. The syntax is a little odd in that the parentheses around the address are required to evaluate the address string back into a numeric value, but it works! Now any time I want to call a method or access a member variable in `OleTCl` I use `loc:Oletcl.Method` or `loc:Oletcl.member` . Back in the program thread where I created `oletcl` I use `oletcl.method` or `oletcl.member`.

Now that `OLEHost` can read and write information in the `oletcl` via the `loc:oletcl` reference variable, I can put it to work creating the hidden window and ole control.

Creating The COM Thread

The code to create the OLE control in OLEHost is fairly straight forward.

```
OLEHost Procedure(String StrAddr)
Window WINDOW('OLEHOST'),AT(,,116,99),GRAY
      END
LongAddr      Long,auto
Loc:OLETCl   &OLETCLType
I long,auto
  code
  !create our class reference.
  LongAddr = strAddr
  loc:OleTCl &= (longAddr)
  !open the window
  Open(window)
  0{prop:hide}='1'
 !Accept loop
 Accept
   Case Event()
   Of Event:OpenWindow
     !create the control, save the Field Equate Label (FEQ)
     loc:OLETCL.OLEFEQ=Create(0,create:OLE)
     loc:OLETCL.OLEFEQ{PROP:Create}=loc:OLETCl.ProgID
     !Save FEQ and thread
     Loc:OLETCL.OLEThread = thread()
   End !case event
  end !accept
  close(WINDOW)
  !clear the thread now that the window is closed.
  Clear(Loc:OLETCL.OLETHREAD)
  Return
```

This procedure creates the COM object and stores the Field Equate Label (FEQ) plus its thread in the `oletcl` class using the reference variable `loc:oletcl` I created earlier. In addition, `OLEHost` gets some information it needs, like the `progid` used with `prop:create`, from the `oletcl` in the same manner. This procedure is located in OLETCL.CLW class file. Please consult the downloadable code supplied with this article to get a better feel for where everything is.

Here's the supporting class code to `Start()` this procedure. The `Init` and `Kill` methods look like this:

```
OLETCLType.Init                    procedure(string pProgID)
DummyWindow WINDOW,AT(,,41,37),timer(1)
      END
StrAddr string(80),auto
  Code
  !save ProgID
  SELF.ProgID    = pProgID
  !convert class address to a string
  StrAddr=address(SELF)
  !start the hidden window with the ole control,
  ! wait for the thread to be set.
  clear(SELF.OLEThread)
  Open(DummyWindow)
  0{prop:hide}=1
  Accept
    Case Event()
    of Event:OpenWindow
        Start(OLEHOST,40000,StrAddr)
    Of Event:Timer
      If SELF.OLEThread then
       0{prop:timer}=0
       Post(Event:CloseWindow)
      End
    end
  end !accept
  close(dummywindow)
  Return

OLETCLType.Kill                    Procedure()
DummyWindow WINDOW,AT(,,41,37),timer(1)
      END
    Code
    !call any kill/Quit methods now
    If SELF.OLEThread then
      Open(dummywindow)
      0{prop:hide}=1
      accept
        Case Event()
        of Event:OpenWindow
          Post(Event:Closewindow,,SELF.OLEThread)
        Of Event:Timer
          If ~SELF.OLEThread then
```

```
                     0{prop:timer}=0
                     Post(Event:CloseWindow)
                  end
             End !case
          end !accept
          Close(DummyWindow)
       end
    Clear(SELF.OLEFEQ)
    RETURN
```

The `Init` method opens a dummy window just so the `Start()` command is executed immediately. `Start()` doesn't happen until an accept loop has cycled. The `Kill` method also opens a dummy window to post the close window event. Both use the value of `SELF.OleThread` to determine when the job is done and it's safe to close the dummy window. If the code for `OLEHost` and the `Init/Kill` methods is compared, I think the idea of how the two pieces communicate through common member variables will be apparent.

Once the OLE control is created by a call to `Oletcl.Init()` I can call the OLE control or OCX's methods. Of course since the OLE control is really on another thread I have to use `SetTarget(,Oletcl.OleThread)` before all calls. That can get a bit tedious, so I added generic class methods to `Oletcl` to call a method, set a property or get a property. For example the code for `Oletcl.callmethod(string methodname)` is:

```
Oletcl.callmethod procedure(string methodname)
  code
  if SELF.OLEThread then
    settarget(,SELF.oleThread)
    SELF.OleFEQ{clip(methodname)}
    settarget()
  elsif SELF.debugmode then
    Message('No OLE for call method on ' |
      & clip(pcmd),'fatal error')
  end
```

There really isn't much there but it relieves the tedium of calling `SetTarget()` and does some error checking.

In late-binding OLE the distinction between a method and a property is blurred. In this context I use method to mean anything that does not return a result where as the `GetProp` method can be used for both properties and methods that return a result.

Some OLE properties and methods require special handling because they return an OLE object rather than the more typical string result. In VB the special handling syntax involves the use of the VB keyword 'Set'. In Clarion no special function or keyword is needed. The object is stored instead in an ordinary cstring(20). All objects stored in a cstring have a ' character as the first character. That fact makes error checking easy. To summarize:

In VB:

```
Dim DrivesObject
Set DrivesObject = FileSystemObject.Drives
```

In Clarion:

```
DrivesObject  cstring(20)
Settarget(,OLEThread)
DrivesObject = OLEFEQ{'DRIVES'}
If DrivesObject[1]<>'`' then
   Message('Get object failed')
end
Settarget()
```

The `GetObj` **method in** `Oletcl` **does the above with a bit of additional error checking.**

In VB when done with an object, it is set to nothing:

```
Set DrivesObject = Nothing
```

In Clarion you use Prop:Release:

```
Settarget(OLEThread)
OleFEQ{prop:Release}=DrivesObj
Settarget()
```

To automate this I also wrote a release object method. Although pretty simple, it does some error testing and determines if the object is valid and needs to be released before actually releasing it. After releasing I clear the object (a cstring) so that if release is called a second time in error, the error is detected and the object is not released twice. However, occasionally I find OLE objects get stuck in memory and do not leave when I think they should. This indicates one or more objects have not been properly released. To remedy that situation, the releaseObj also has a count parameter that defaults to 1. If greater than one, Prop:Release is called multiple times before the object (cstring) is cleared. Lastly, I wrote a `CallObj` method to allow calling a property or method on an object obtained with `GetObj`.

One last difference between the VB code and Clarion code deserves comment.

In VB:

```
Set DriveObject = DrivesObject("C")
```

In Clarion:

```
Oletcl.GetObj(DrivesObject &'.item("C"))
```

The `DrivesObject` is a special sort of object called a collection. A collection can be thought of as a queue of OLE objects. In this case, the collection contains `DriveObjects`. All collection objects have three properties/methods: `count`, `item`, and `_NewEnum`. The count property returns the number of items in the collection. The `item()` method is used to get an individual member of the collection. Most commonly, the item method takes an integer in the range $1...Count$. But rules were made to be broken, and other collections take $0...Count-1$ as a parameter. In still other cases, the item method takes a key value with a special meaning within the context of that collection. Here the item method takes a key value in the form of a drive letter. There is no way to know without consulting a utility program or documentation how a particular collection works. I usually try $Item(1...Count)$, and if that fails I just try the other methods until something works. I suppose I could read the help but where's the sport in that?

Those who have read my other articles on early binding may be interested in the

_NewEnum method, which VB uses. Visual C on the other hand does not support collections and Visual C programmers like Clarion programmers most often use the Item()/Count method. When VB calls the _NewEnum method an IEnumVariant interface is returned. That interface has the methods expected for enumerating a collection like Next, reset, and skip (although no previous). By obtaining an IEnumVariant interface and calling the Next method repeatedly, it is possible to loop through the entire collection without knowing the key values. Calling _NewEnum and IEnumVariant both require early binding techniques (discussed in my earlier articles) which are more difficult, but still possible in Clarion. If for some reason it is not possible to know the key values required, then it's time to dig in and use early binding to call _newenum and the necessary methods on the returned IEnumVariant interface. An example where this might be necessary is a collection of license plate numbers that uses key values for retrieval. Since there are many possible values and they don't fall in an easy-to-discover sequence, enumeration using IEnumVariant may be the best way.

To demonstrate more of the methods of Oletcl class not used in the file system object above look at some code to put a value into cell (1,1) on a new Excel spread sheet, and save it to disk in a new file called c:\testsave.xls:

```
cXlSname='C:\testsave.xls'
remove(cXLSname)
oletcl.init('excel.application')
oletcl.callmethod('workbooks.add')
oletcl.setprop('cells(1,1).value','test')
oletcl.callmethod('activeworkbook.saveas("'&cXlsname&'")')
oletcl.callmethod('quit()')
oletcl.kill()
```

As you would expect the init method creates the excel.application object on the hidden thread. Next the callmethod method is used to add a new workbook. Once the workbook is created it becomes the default workbook and don't have to specify which workbook I'm using in subsequent calls. To set the value in cell 1,1 I use the setprop method to set the value property for the cell. Since the OLE code deep inside Clarion will convert most parameters to a special COM type called a variant, even if the property you want to set is a long, passing it as a string does no harm.

The next two lines use the callmethod method again to save the new spreadsheet and exit excel. When I compiled and ran the code, it did its job and saved the new spreadsheet, but watching in NT task manager, after the quit() and up until my application ended, Excel was still in memory. No matter how long I waited after the quit(), Excel was still there. At first I though it might have to do with the fact that workbooks.add does return a workbook object that I am ignoring but even accounting for that and releasing the workbook object, Excel was still stuck in memory until my application ended. Since it looked to me like it was the application object that was stuck in memory, I decided to get the application object and release it myself until Excel bit the dust. The code that finally worked was this:

```
cXlSname='C:\testsave.xls'
remove(cXLSname)
oletcl.init('excel.application')
oletcl.getobj('application',capplication)
oletcl.callmethod('workbooks.add')
oletcl.setprop('cells(1,1).value','test')
oletcl.callmethod('activeworkbook.saveas("'&cXlsname&'")')
oletcl.callmethod('quit()')
oletcl.releaseobj(capplication,3)
oletcl.kill()
```

Notice the two extra lines, one to obtain the application object and store it in `capplication,` and the next-to-last line to release it three times. Why three rather than the expected one time I have no idea, but it works. Following the three releases, Excel is gone from memory as shown in the task manager. There are many cases where it may be advantageous to keep Excel in memory while until the program exits, but if that isn't desirable, release will give it the boot.

That completes the `Oletcl` class. The downloadable code (see below) has a few enhancements for debugging, and allowing multiple OLE objects (up to five) to be put on one window. The OleTest.prj contains the sample code used in the article. Using the `OLETcl` class I can rewrite the VB code line for line. But considering all the "constructive criticism" (a.k.a. whining) on the newsgroups about Clarion's COM support, the code is remarkably similar.

What ever happens do please don't show this to a VB programmer. If we take away ease of use of COM as an advantage, what will they have left to hang on to?

[Download the source](#)

Jim Kane was not born any where near a log cabin. In fact he was born in New York City. After attending college at New York University, he went on to dental school at Harvard University. Troubled by vast numbers of unpaid bills, he accepted a U.S. Air Force Scholarship for dental school, and since graduating has served in the US Air Force. He is currently the Officer in Charge of Dental Facility Design at USAF Dental Investigation Service in San Antonio, Texas. In his spare time, he runs a computer consulting service, Productive Software Solutions, which he hopes to run full time after retiring from the US Air Force Dental Corps in June 2000. He is married to the former Jane Callahan of Cando, North Dakota. Jim and Jane have two children, Thomas and Amy.

Reborn Free   CLARION online   published by CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

The Clarion Challenge: Clarion COM Scripting

In this week's issue Jim Kane explains how Clarion COM can be as easy as VBScript COM. And why would that matter? Well as it turns out, VBScript is pretty powerful. You can use it to search for files, write data, send emails, and more, as everyone who's seen the I LOVE YOU virus can attest. Of course, that doesn't mean you should start writing viri in Clarion (use the Force for good, Luke).

This Clarion challenge is all about creativity. Given the power of COM and the tools provided by Jim Kane, what kind of Clarionized VBScript can you create?

Send your entries to editor@clarionmag.com. We'll post the best submissions in an upcoming issue.

Clarion COM: As Easy As Visual Basic COM
(May 9,2000)

Clarion Challenge: COM Examples
(May 9,2000)

Cloning List Boxes - Source Update
(May 9,2000)

Cloning List Boxes

By Jon Waterhouse

Part 2 of 2

There are many occasions when it is useful to have list boxes that are clones of each other. In some cases these will be created by the developer, as in an appointment manager, where the screen contains a month's worth of identical boxes showing appointments for the day. Other times these will be created at runtime, as in a program where the user is classifying things into an unknown number of categories (say, students into classes, creating extra classes to hold students as necessary).

Last week I introduced the concept of the cloneable list box and looked at some of the template requirements. This week I'll conclude by covering the template code in detail.

As an aid to understanding what needs to go in the template, here is an outline of the code the template needs to produce (the *5* in MBQ5 is determined by the sequence number of the control on the window):

Figure 1: Outline of code to be generated.

```
MBQ5Type          QUEUE,TYPE    !Queue type declaration
!Things in italics are pseudo code
Normal queue element declarations
                  END


MBQ5              &MBQ5Type     !Ref to queue type
NewControl3       NewControl

! Multi box manager
MB5               CLASS(MultiboxClass2)
!Reference to display queue
Q                 &MBQ5Type
AddBox            PROCEDURE()
Kill              PROCEDURE(),DERIVED
TakeEvent         PROCEDURE(Signed control),BYTE,DERIVED
TakeNewSelection  PROCEDURE(),DERIVED
```

```
                     END

MB5_ref          &MB5              !Reference to Multibox class

    CODE
    ...
MB5.AddBox PROCEDURE
    CODE
    ! Create new class instance
    MB5_ref &= NEW MB5
    ! Create new queue for new class instance
    MBQ5 &= New MBQ5Type
    ...
    MB5_ref.Init(init parameters)
    MB5_ref.Q &= MBQ
    MB5_ref.AddSortOrder(sort parameter)
    ...
    MB5_ref.ListFEQ{PROP:Format}=format string
    MB5_ref.ListFEQ{PROP:DropID}=dropID
    MB5_ref.ListFEQ{PROP:DragID}=dragID
    !Fill the queue
    MB5_ref.ResetQueue(1)
    ...
    NewControl3.AddControl(MB5_ref,MB5_ref.ListFEQ,3)
     ...
    END
```

You may have noticed that the `AddBox` method has no references to `SELF` or `PARENT`. The method could, with just as much validity, be declared as a local `WindowManager` method. The main reason for I set it up as a `Multibox` method is that it is much easier to make a derived method for the class that the template is dealing with in any case, than it is to add a method to some other class.

The best way to proceed from here is to look at the template itself. Only the segments of the template that are changed substantially from the original FileDrop template, or are of interest for some other reason, are given here. Each segment is preceded by an explanation of the code.

The `#CONTROL` statement says that the control requires a primary file (with a required key), that it requires a window, that multiple copies are allowed, that this control template will be offered as a choice whenever you add a naked droplist to the window (the `WRAP` option), and that it requires the `NewControl` template to be present.

```
#CONTROL(Multibox,'Multi Box'),PRIMARY('Multi Box'),↵
   DESCRIPTION('Multi Box on ' & %Primary),WINDOW,MULTI,↵

HLP('~TPLControlFileDrop'),WRAP(DropList),REQ(NewControl)
```

In the `#PREPARE` section, the `%SetOOPDefaults` requires that the `%MultiboxType` token exist, and this is why I have declared it at the start of the section. The Application Generator gives you the choice of using the default ABC class associated with each template, or nominating a new one. In the AppGen this is accomplished for the ABC classes by `#PROMPT` statements found on the Classes tab of Global Properties. The template code

that produces this tab is found in the ABCHAIN.TPL file. If desired, you *could* add to the code in ABCHAIN.TPL and allow programmers to nominate their own classes to associate with the `NewControl` and `Multibox` templates. For now, I want to keep the changes made to other files to a minimum, so the `#EQUATE` (which is the same as `#DECLARE` and `#SET`) fulfills the same purpose of instantiating the `%MultiboxType` token.

```
#PREPARE
  #EQUATE(%MultiboxType,'MultiboxClass2')
  #CALL(%ReadABCFiles)
  #CALL(%SetClassItem,'Default')
  #CALL(%SetOOPDefaults,'MB'&%ActiveTemplateInstance,↵
     %MultiboxType)
#ENDPREPARE
```

Next, the `CONTROLS` statement specifies the controls to be added to the window. The `HIDE()` and `FROM()` attributes are added here because a) the control added in the window formatter should remain hidden, and b) the queue is being declared as a `TYPE` and the default behaviour of the FileDrop template would be to use the queue type as if it was a real queue. This would cause the program to GPF when it hit the OPEN(Window) statement, where the window definition includes LIST,FROM(AQueueTypeNotARealQueue). The next section of the template contains the property sheets that the programmer fills in to make the Multiboxes behave in the desired manner. Much of this is lifted directly from the ABDROPS.TPW file. However, the section to do with the update targets on acceptance is deleted, and a tab with items specific to the Multibox operation is added. It is worth mentioning here that the `%OOPHiddenPrompts` section is extremely important, and cannot be omitted. This group, which is contained in the ABGROUP.TPW file, uses `#PROMPT` statements to instantiate four tokens, two of them multi-valued, that allow the template to construct the class definition. They are:

- `%Classitem`: which is a multi-valued symbol that will hold the classes that need to be declared in the module
- `%DefaultBaseClassType`, which is taken from the `%MultiboxType` token that was declared in the `#PREPARE` section
- `%ActualDefaultBaseClassType`: which is set in code and is used in the ABC classes to select the type of step manager to use
- `%Classlines`: which is a multi-valued symbol that will contain the lines to place in the class declaration, including both properties and methods. In particular, items get added to `%Classlines` as the code generator encounters code placed in embeds associated with class methods.

The prompts on the *Special Multibox* page include places for the programmer to define:

- The field that provides the range limit value on initialization
- The field that provides the box title on initialization
- The component of the primary key that will be used in the range limit

The range limit has to be (in this implementation) a single value, and the field that will be used as the limit is `SELF.LimitValue`, one of the properties specific to the Multibox2 class.

Figure 2: Programmer input.

```
#BUTTON('File Drop Properties')
   #BOXED('Default Resizer prompts'),AT(0,0),↵
         WHERE(%False),HIDE
      #INSERT(%OOPHiddenPrompts)
   #ENDBOXED
   #SHEET,HSCROLL
      …
      #TAB('&Special Multibox stuff'),WHERE(%PrimaryKey)
       #PROMPT('&Region where box can be dropped:',CONTROL),↵
          %MoveRegion,REQ
       #PROMPT('Limiting field for Multibox:',FIELD),↵
          %LimitField,REQ
       #PROMPT('Title string for Multibox:',FIELD),↵
          %TitleField,REQ
       #PROMPT('Range limit &field:',↵
           COMPONENT(%PrimaryKey)),%RangeField
       #ENABLE(%RangeField)
         #PROMPT('Range limit &type:',DROP('Single Value')),↵
            %RangeLimitType,DEFAULT('Single Value')
         #BOXED('Range limit boundary'),↵
             WHERE(%RangeLimitType='Single Value'),AT(,85)
           #PROMPT('&Range limit value:',@s20),↵
              %RangeLimit,DEFAULT('SELF.LimitValue')
         #ENDBOXED
       #ENDENABLE
      #ENDTAB
      …
      #TAB('Cl&asses')
        #WITH(%ClassItem,'MultiboxClass2')
          #INSERT(%ClassPrompts)
        #ENDWITH
      #ENDTAB
   #ENDSHEET
#ENDBUTTON
```

The last tab is in the prompts section is the classes tab, which is displayed by the `%ClassPrompts` group contained in ABGROUP.TPW. The call to `%SetOOPDefaults` in the #PREPARE section will already have set the name of the object to `MBx`, and specified that the object will use the ABC default Multibox2 class.

The next section (`#AT(%AtCustomGlobalDeclarations)`) is unchanged from the FileDrop template. It is important because, in addition to preparing a list of icons to merge into the executable, it also calls the `%FileControlSetFlags` GROUP, which is the group that adds the primary and related files of the list box to the list of files that need to be declared globally.

The Tricky Bit

To understand the next part of the template you will probably find it useful to refer back to the outline generated code listing in Figure 1 of Part 1.

First the class is declared, followed by a reference to that class. Each time a new Multibox is

added, the `MB5_ref` variable will reference the new class, and the `MB5_ref` reference will also be the reference stored in the queue in `NewControl`. The method declarations will be declared as `MB5.Addbox`, for example, but any method calls will be to `MB5_ref.methodname()`. Some of the groups that are called use the `%FileDropObjectName` token, and this must also be set to the reference to the class, and not the class itself, so that the code is correctly generated. This is accomplished in the declaration of the `%Mbref` token, and the setting of the `%FileDropObjectName` to `%Mbref`. The `%AtWidth` and `%Atheight` tokens are added because the `Init` method needs to take the control's height and width as parameters.

At the end of this section, the `%FileDropAtStart` group does some useful stuff in initialising tokens. However, one of the things it does is to set the `%ListQueue` token to the `FROM()` attribute of the control (which should be always be blank in the generated code). The way the queue is dealt with has similarities to the class reference method; the queue type is declared, there is a reference to the queue, and this reference is what is used by the class. The last two lines therefore declare the two tokens for the queue type and the queue reference variable, overwriting what has already been set up in the `%FileDropAtStart` GROUP.

Figure 3: The #ATSTART section.

```
#ATSTART
  ...
  #DECLARE(%AtWidth)
  #DECLARE(%AtHeight)
  #EQUATE(%MBref,%ThisObjectName&'_ref')
  #EQUATE(%FileDropObjectName,%MBref)
  ...
  #EQUATE(%InstancePrefix,'MB'&%ActiveTemplateInstance&':')
  #EQUATE(%ListView,%InstancePrefix&':View:FileDrop')
  #INSERT(%FileDropAtStart)
    #! The listqueue token is set in FileDropAtStart
  #SET(%ListQueue,'MBQ'&&%ActiveTemplateInstance)
  #EQUATE(%ListQueueType,%ListQueue&'Type')
#ENDAT
```

The `%GatherObjects` and `%DataSection` segments of code remain unchanged from the file drop template. The latter contains the call to construct the `VIEW` definition. The next few lines of code are also unchanged, but it is probably useful to point out that it is this little bit of code that produces the `Parent.WhateverMethod()` code in the middle of each derived class method.

```
#AT(%FileDropMethodCodeSection,%ActiveTemplateInstance)↵
    ,PRIORITY(5000),DESCRIPTION('Parent Call')↵
    ,WHERE(%ParentCallValid())
  #CALL(%GenerateParentCall)
#ENDAT
```

The next section of code lays out the items that will be included in the `Addbox()` method of the class. The `Addbox` method is declared in the class .inc and .clw files, but the default method does not do anything. Basically, it just adds a place to easily derive a local class method, the same way, for example, that the `ValidateRecord` method in the process

template does. In theory it should be possible to add any number of new derived methods using the template, with additions to the `%ClassMethods` token that would mimic adding a method using the New Class Method button on the Actions|Classes tab. In practice it is easier to add a stub method to the ABC class and just provide the code for the appropriate embed in the template to produce the local derived class. This is how the `WindowManager` works with, for example, the `WindowManager.TakeFieldEvent` method doing nothing itself, but relying on the template to produce a locally-derived class that does what is required.

With the method already declared in the class file, producing a derived method is absurdly easy. Just having code in an `#AT` that references the embed point in the method gets the method declared in the class declaration.

The new class instance is created and is referred to by the `%FileDropObjectName` token. The `%Atwidth` and `%Atheight` tokens are constructed by `EXTRACT`ing the attributes from the control that belongs to the template. The `#SET` after the `Init` is enclosed in an `#IF` statement because it doesn't work otherwise. For some reason, the `%RangeLimit` is undefined when the `#IF` statement is not used.

Note that method calls are prefixed by `%FileDropObjectName` and not by `SELF`, because it is the new instance of the class that must be acted on, not the parent. The `%FileDropAfterWindowOpening` GROUP does a lot of the work in this method; it produces all the `AddField()` calls.

The `FORMAT` statement is `EXRACT`ed from the `%ControlStatement` so that it can be applied to each newly created list box. The `DROPID` and `DRAGID`, if they exist, are also copied to each of the new list boxes.

The `%GetNewControlManager` GROUP is necessary because the name of the `NewControl` class instance is needed so that the class can be given the new `Multibox` to manage. The GROUP has the `PRESERVE` attribute, so that after returning the `%ActiveTemplate` token still refers to the `Multibox` template, and not to the `NewControl` template.

Once everything has been initialised, the new controls that make up the Multibox are unhidden.

Figure 4: The AddBox template code.

```
#AT(%FileDropMethodCodeSection,%ActiveTemplateInstance↵
   ,'AddBox','(SIGNED xpos,SIGNED ypos)')
%FileDropObjectName &= NEW %ThisObjectName
 %ListQueue &= New %ListQueueType
 #FIX(%Control,%DropControl)
 #SET(%AtWidth,EXTRACT(%ControlStatement,'AT',3))
 #SET(%Atheight,EXTRACT(%ControlStatement,'AT',4))
 %FileDropObjectName.Init(%LimitField,%MoveRegion↵
     ,%TitleField,xpos,ypos,%Atwidth,%Atheight↵
     ,%ListQueue.ViewPosition,%ListQueue,%ListView↵
     ,Relate:%Primary,%WindowManagerObject)
 #IF(%RangeField)
   #SET(%RangeLimit,%FileDropObjectName&'.LimitValue')
 #ENDIF
 #INSERT(%FileDropAfterWindowOpening)
```

```
  #SET(%ValueConstruct,EXTRACT(%ControlStatement,'FORMAT',0))
  %FileDropObjectName.ListFEQ{PROP:Format}=%ValueConstruct
  #SET(%ValueConstruct,EXTRACT(%ControlStatement,'DROPID',0))
  #IF(%ValueConstruct)
  %FileDropObjectName.ListFEQ{PROP:DropID}=%ValueConstruct
  #ENDIF
  #SET(%ValueConstruct,EXTRACT(%ControlStatement,'DRAGID',0))
  #IF(%ValueConstruct)
  %FileDropObjectName.ListFEQ{PROP:DragID}=%ValueConstruct
  #ENDIF
  %FileDropObjectName.ResetQueue(1)
  #SET(%ValueConstruct,%WindowManagerObject↵
    &'.additem('&%FileDropObjectName&')')
  SETCURSOR(CURSOR:Arrow)
  disable(%MoveRegion)
  #INSERT(%GetNewControlManager)
  %ValueConstruct.AddControl(%FileDropObjectName↵
    ,%FileDropObjectName.ListFEQ,3)
  UNHIDE(%FileDropObjectName.ListFEQ)
  UNHIDE(%FileDropObjectName.LabelFEQ)
  UNHIDE(%FileDropObjectName.RegionFEQ)
 #ENDAT
```

The Kill method needs to DISPOSE the queue (which is declared locally) before calling the parent kill method that DESTROYs the controls that were created by Addbox.

```
#AT(%FileDropMethodCodeSection,%ActiveTemplateInstance↵
   ,'Kill','()')
DISPOSE(SELF.Q)
Parent.Kill
#ENDAT
```

The code in #AT(%LocalProcedures) is almost unchanged from the code in the FileDrop template. The only difference is that the Q in the class declaration needs to reference the queue type, rather than a real queue. The code here also performs the magic that adds values to the %ClassLines token if there is code added to an embed point (either by the template or added by the programmer). This then results in the methods being added to the class declaration by the %GenerateClassDefinition GROUP. The embed points here are still indicated as %FileDropMethodCodeSection, and this matches how the #ATs have been defined earlier. It would be possible to change the embed point name to %MultiboxMethodCodeSection, but this would mean that any code from the FileDrop template would have to be copied here. For example, getting conditional colours to work would require copying over (and changing the embed name of) the SetQueueRecord section from ABDROPS.TPW. There is little advantage in doing this since keeping the same name causes no confusion. The %ActiveTemplateInstance token will ensure that the code still gets generated where it is supposed to.

The #AT(%LocalClassData) section has to be changed because it calls the %ConstructQueue GROUP, which adds the queue declaration in the data section. This is changed to %ConstructQueueType, as discussed below.

The %LocalDataClasses section has the call to %GenerateClassDefinition

which actually writes the class declaration based on the contents of the `%ClassLines` token. The `%GenerateClassDefinition` GROUP also calls `%AddModuleIncludeFile` which makes sure that the .inc file of the referenced class or classes is included into the module. Following this the `%OOPConstuct` is made which produces the reference to the class, i.e. MB5_ref &MB5.

Figure 5: #AT(%LocalDataClasses)

```
#AT(%LocalDataClasses)
#CALL(%SetClassItem,'Default')
#INSERT(%GenerateClassDefinition,%ClassLines↵
    ,'Multi box manager')
#INSERT(%MakeDeclr,24,%OOPconstruct,%FileDropObjectName↵
    ,'&'&%ThisObjectName)
%OOPconstruct
#ENDAT
```

The `%ConstructQueue` GROUP is over 50 lines of code, and unfortunately all of this needs to be copied into the template event though only minor modifications are required. These are to add the `,TYPE` to the queue declaration, and add the queue reference after the declaration. The modified version is called `%ConstructQueueType`.

Finally, the `%GetNewControlManager` group fixes the `%ActiveTemplateInstance` to NewControl(ABC), and loops through the `%ActiveTemplateInstance` to find the instance of the class (there should be only one).

That completes the `Multibox` control template. The `NewControl` template is an `EXTENSION` template that wraps the `NewControl` class. The template is quite simple. What it accomplishes is the following:

- Takes note of the last regular field on the window when it is opened; any FEQs greater than this will be handled by the `NewControl` class
- The `WindowManager.Kill` method also has to call the `NewControl.Kill` method, which in turn kills the Multiboxes that have been added
- In the Window Manager's `TakeFieldEvent` method the `NewControl.TakeEvent` method is called for the newly-added controls.

The example application is based on two files, SHOOT and SCENE. It is part of a program for scheduling a movie shoot. This part of the program is for working out which groups of scenes will be shot together on one day. The following few lines of code are all that is required to add lists for all the existing shoot days to the screen:

Figure 6: Code to add the Multiboxes.

```
x_pos=?region1{PROP:At,1}
y_pos=?region1{PROP:At,2}
set(shoot)
loop
  if access:shoot.next() then break.
  mb4.addbox(x_pos,y_pos)
  x_pos+=80
  if x_pos > ?region1{PROP:At,1}+?region1{PROP:At,3}-60
     x_pos=?region1{PROP:At,1}
     y_pos+=80
     if y_pos > ?region1{PROP:At,2}+?region1{PROP:At,4}-60 ↵
   then break.
  end
end
```

To run the demo you will need to:

- Copy the .clw and .inc files into your libsrc directory
- Copy the .tpw file into your template directory, and make the template file part of your template set by adding this line

```
#INCLUDE('ABMULTI.TPW')
```

to your ABCHAIN.TPL file (about line 590)

After that, you should be able to load and compile the mbdemo.app. This app creates four clones of the original list box, but if you wish you can easily give your users this kind of control, letting them create and arrange new lists as needed.

[Download the source code](#)

[Jon Waterhouse](#) has been using Clarion since the 2.1 days. His main work is as an economist, and he finds that Clarion is well-suited for applications which impose order on various sets of data. His projects include questionnaire data entry programs, classification software (assigning projects to groups), plus some more interesting scheduling applications. Jon has also used Clarion to link text information together, and is currently developing a program that will store linked snippets of WordPerfect documents and print custom documents composed of several of these snippets. He is currently working for the Newfoundland Government on a project to measure the performance of government employment programs.

Reborn Free　　CLARION online　　published by CoveComm Inc.

Clarion MAGAZINE

Interview: SoftVelocity's Bob Zaunere

On May 2, 2000 Frank Watts, TopSpeed's President and CEO, announced the formation of two separate, independent companies to continue TopSpeed's business directions. One of these is SoftVelocity Inc., which takes over TopSpeed's Clarion line of rapid application development tools. SoftVelocity's President and CEO Robert Zaunere spoke with Dave Harms, Clarion Magazine's editor and publisher.

Let me just add my congratulations to everyone else's. I think we're all happy to see someone with your background and interest in Clarion take over.

I'm really happy about it. For a long time I've listened to input from our user community, I've heard about the problems as they are, and I've often wished I could do something to have immediate impact. And now I'm in that position.

What made you put your offer on the table?

I believe in the product and I believe in its potential, especially its untapped potential. So when the opportunity was presented to me, I had to do it. There are many things that I felt were needed in the product, based on my experience and input that I've had from users over a long number of years. And the opportunity to have the freedom to pursue that was irresistible.

Who's behind SoftVelocity?

It's a small group of investors who all believe, as I do, that Clarion has a unique position in the market and a long life ahead of it.

Is Bruce Barrington still involved?

While Bruce isn't going to be designing Clarion on a day to day basis, we are working together on the transition, and he is mentoring me when I need it.

Will TopSpeed cease to exist?

TopSpeed is going through the process of liquidation of its assets. That will probably take a couple of years.

TopSpeed

Clarion 5
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

Will you be retaining the TopSpeed web site?

Bruce and I were just talking about that. I don't know that we'll retain the TopSpeed name long term. For the next six to nine months it'll be ours to use. But I don't know that we need it. As soon as the dust settles we'll be finishing up our new web site, and we'll have redirection from the old web site for a long time.

There's a lot of curiosity about how the splitting of TopSpeed into two companies came about. I understand there has been an internal division in the company for a while between desktop and web development.

There have been two separate business units. We've had the Clarion core business and we've also had the Wired On MainStreet initiative, which is really about Internet marketing. Wired On MainStreet's products and services are targeted at non-technical people, at the business community and not the product development community. They use Clarion, as you would expect, as the cornerstone of their back office operations.

It seems like their content delivery is based on Macromedia's Flash.

They have a couple of guys who are big believers in Flash.

So they're not using iBuild?

Not so much to deliver content, but they are using the messaging capabilities [of C5.5]. They have an email campaign manager product, which is 95% complete, written entirely in 5.5. It's a pretty sophisticated piece of technology that has connectivity to web databases as well as a back office database, and does synchronization between them.

Is that something you might license for the Clarion product line?

No, that is their product entirely. We're in the tool business.

Is the iBuild dream over?

We were trying to create a bridge between two separate communities, and iBuild was not really going to make it. These communities are distinctly different, and I came to believe that it couldn't be done.

Will Sensium be doing any iBuild development?

They do not develop anything that has to do with the tool business. They will use Clarion to develop applications.

Will the iBuild technology rolled into 5.5 still be developed?

The components of iBuild that proved themselves viable through the 5.5 beta cycle will be developed and expanded on. Certainly all the messaging support, the technology dependent on the Winsock layer are included. The idea of the Wizatrons creating web apps is probably not one we're going to put as much emphasis on as was previously indicated.

We're going to begin by focusing on our strength as a desktop development tool. We'll enhance our ability to make use of the internet from within Clarion applications. No one can afford to ignore the web, but on the other hand the desktop isn't going to disappear.

What does that mean for product development?

Our immediate mission is to polish 5.5 and deliver that technology to our user base. In

the immediate future we want to provide better integration of existing technology like COM and DCOM, as well as emerging technologies like XML and XSL. There will definitely be more development of web technologies.

Where are things at with COM right now?

We've made some important improvements and fixes to the 5.5 COM layer, just within the last few alpha builds. We need to go a lot further with it, but I think we've taken an important step with 5.5, and my plan is not to wait until a full release later but to find a way to incrementally deliver COM and DCOM.

But the app broker/ASP market isn't a focus?

The app broker is a very useful technology, and there are lots of times when it makes sense, but it doesn't make sense for every application, and I don't think that it has always been positioned correctly.

Where does it best fit?

For the Application Server Provider market, the app broker technology has some distinct advantages over other approaches. Expect to see more development in this area. For instance, the app broker technology is ideal for business to business and extranet applications. The ability to encapsulate business logic, which is Clarion's strength, and deliver it over the Internet, is a powerful feature. It's a very unique technology, but you have to pick the tool that is best for the job. It is best for some jobs, but I don't believe that it's a general purpose tool.

So it fits in, say, somewhere between Citrix's MetaFrame and a regular web server?

That's an excellent way to put it. There are a lot of people who have found that for their application it is the ideal solution, and they're amazed at how quickly they can encapsulate their business logic and deliver it over the web. But I wouldn't say let's go out and build amazon.com using it.

Do you have any plans to generate code for other languages.

For a lot of years I've said there are other languages that we can generate code for. My idea about that hasn't changed at all. If anything I feel even more strongly that we should take advantage of that strength.

Is Java one of those languages?
Java is definitely in the cards.

Richard Chapman at DevCon said that Clarion was TopSpeed's strategic language for client/server development, and Java was their strategic language for distributed multi-tier development.

What Richard said about Java is accurate, but we haven't completely decided where we want to go with it. We need more input from our user community. We've only been in business ten days so we need time before our strategy for Java takes shape. I certainly believe in Java. I like the language; I've been using it since the first release of the JDK. It has its place.

Are there still people in London doing Java development?

Absolutely. They've added a couple of new people for just that purpose.

What kind of work are they doing?

I can't say what kind of work they are doing at this time. Both the LDC and I are under a non-disclosure agreement.

Are there any changes coming up in calling conventions, threading, any of the areas where Clarion isn't in line with industry standards, de facto or otherwise?

I'm a firm believer in industry standards, de facto or otherwise. So yes, we'll definitely be moving in that direction.

How about C6?

C6 has been a work in progress for a long time, and a lot of what you see in 5.5 was originally slated for C6. I would like to have a firmer specification of what goes into C6 and in order to do so I'm going to look for a lot more input from our user community. The survey we did was a step in the right direction, but it only scratches the surface.

Will there be a 32 bit IDE in C6?

It's an absolute must. I'll leave it at that.

Do you have any plans for Linux?

Absolutely. It's hard to ignore Linux. Right now it's very strong on the server, both here and in Europe, and we are currently looking at options for using Clarion on the Windows desktop connecting with back end databases on Linux servers.

Which we can do now with ODBC.

Well, there are a couple of other things that I'm not prepared to talk about right now that we're looking into. It's also another area that we might be able to help our development community with education and technical support. We've been running Linux here for quite a few years so we have quite a bit if in-house expertise, and we'll probably bring in some additional people as we see the need for it.

Who owns the London Development Centre?

The DC has been and still is owned by TopSpeed Corporation.

So TopSpeed has always worked with the DC on a contractual basis?

The DC has operated as a separate entity from TopSpeed Corporation.

There have been rumors and fears that if another company bought the DC it could cripple TopSpeed's (and now SoftVelocity's) ability to deliver product.

Our ability to deliver product has been enhanced. Certainly we have the contract with the Development Centre and we can continue to get their services. But the contract doesn't restrict us to only using the DC. I'm in contact with some of the developers who have left the DC, and we're going to be working with them. I've also been speaking with some of the original developers of Clarion 2.1. We can hire or contract with people who can get the job done. I'm very optimistic that we'll be able to do more in a shorter period of time than was previously possible.

There seems to have been a high level of attrition at TopSpeed in recent months. Is that an accurate perception?

This industry is known for a high degree of mobility in the work force. The

Development Centre has had some people leave, and they've also added some additional staff. The same is true of us. We've had some people leave, and some of them weren't too happy about that, but this has had no effect on our ability to conduct business, and I think we were in need of some new talent and new energy.

Will the change in ownership have any effect on third party developers?

One of our first priorities was to talk to all the third party developers who are actually our partners. I feel confident we can do a lot more for third party developers. Kelly [Zaunere] is now back in charge of the third party program. She has a very strong line of communication with them, and I think we'll find ourselves working more closely with third party developers than we ever have before.

Who's now doing Clarion development? We haven't seen much of Richard Chapman and David Bayliss lately.

David was primarily responsible for the compiler except when he was working on the Wizatron project. He still will make any changes as we need them for the compiler. Richard was really the managerial lead person, so he didn't have a coding role for Clarion.

Who are the key people right now?

Nigel Hicks is the team leader. Nigel has been with JPI and TopSpeed since the very beginning, from the original Borland team.

And you go back quite a ways too.

Yes, I do. I was hired just over ten years ago, when Clarion was trying to get CPD 2.1 out the door. They needed someone to help clean it up and get it shipping. It had dragged on for quite a while. Funny how things come full circle. Now I'm cleaning up 5.5 and getting that out the door.

What did you do before Clarion?

When I first moved out here I did consulting work, C programming for a company that sold packages to attorneys.

And at TopSpeed you've had a variety of roles. How has that prepared you for this job?

I was also responsible for doing all analysis of products that we were competing against. In doing so I pretty much got my hands on every product that came out that was in the same market space as [Clarion]. That's part of the reason I've stayed here so long. I think every tool has its strengths, and every tool has its weaknesses. Clarion is very good at creating certain types of applications. That isn't to say that I think that it's better in every regard.

How does that affect your marketing?

We'll position the product not so that it tries to take over as much territory as possible, but so that it claims a small area and does it very very well. I believe we need to mesh with the rest of the industry instead of fighting over which tool is better.

Do you have a marketing plan in place or is that under development?

It is under development. Ten days of operations, I guess that includes weekends, hasn't

really given us time to formulate a complete strategy. Certainly we have great appeal to independent software developers. It's been our strength for a long time.

Part of our problem with penetrating the corporate market has been trying to compete directly against other tools that are already entrenched, as opposed to working in a complementary fashion. I have no problem with demonstrating ways to call Clarion DLLs from within Visual Basic, or ways to call Visual Basic from within Clarion. If we try to show how we can work with [other tools] and help larger or smaller companies to complete whatever their mission is, we will do a lot better. My message has always been the same. Don't sell against them. Sell with them. There's no reason that we can't work side by side with other tools.

And you don't object to Clarion developers picking up other tools?

I believe it's probably more common than not that developers have more than one tool.

Who's on your sales team?

Almost all the sales team that worked for TopSpeed, aside from people we didn't see a role for in our new company, are in contact with our customer base now. Our operations manager, Jeff Trudel, has been with the company 11 or 12 years.

Will all development still be done in London?

Our plan at this point is to bring some people in to do development in the Florida office.

How about Jim Defabia? There's a grassroots movement to keep him working on Clarion even though he's moving to Tampa.

Jim will always be involved in the Clarion effort. Certainly he intends to continue to write Dr. Dephobia columns, and he's working on some utility programs which he's offered for us to include in Clarion in the future.

The consulting division has gone to Sensium. Do you know what services they will be offering?

They have contractual obligations for Clarion jobs that they need to fulfill. Their intent, as I understand it, is to move steadily toward doing just web/internet development.

In the past consulting has been seen as a division that kept the company going. Does the loss of consulting revenue worry you?

I'm not concerned about the loss of revenue because we've been profitable based strictly on product sales for a while now. Once we get 5.5 shipping and we're back on track, I'm confident we'll see a positive impact on sales.

I think there are two areas we haven't done as well as we should have. One of those is the support side. We haven't really put the emphasis on delivering support the way that we could have and the way our community wishes we had. The same goes for education. That was allowed to wane away, and I think there's a lot more we can do in that regard. Support and education are critical to helping our developer community, and they'll be an important part of our revenue stream, no question about it.

Will there be changes to the free newsgroups?

We haven't completely decided. My feelings are that the newsgroups as a means for peer-to-peer support are great. But having said that, I think there are a lot of people who avoid them for two reasons. One, there's a very high degree of noise. It's not easy to get

the technical information you need without wading through a lot of non-technical or not useful information. I think there's a large group of developers who would prefer to have a venue under which they could get good quality technical support. That doesn't mean there isn't still a need for people to give free support, but at the same time we have to recognize that a lot of the support that goes on there doesn't help our cause at all. A lot of it actually scares away potential users when they read a lot of negative comments.

Within a couple of weeks I'm going to be finalizing plans for a revamped and re-energized support mechanism. We had the best intentions when we created [the newsgroups], but I don't know of anyone who can get timely information from the newsgroups, or if they do it's hit or miss. And I don't think that's acceptable in a development community.

So you're looking at a tech support service that supplements the newsgroups?

Yes, and I would say the newsgroups will be pared down to something more manageable.

TopSpeed at one time was looking at going public. Is that something you think about.

It's not in the immediate mission plan, however I can't rule it out for some time in the future.

Is DevCon still on for September.

September is coming up very quickly. We haven't officially made a determination on what we're going to do about DevCon, but unofficially I'm inclined to think that we will postpone and try to schedule something for the last part of the year, late November or early December. We'll know better in a week or two, and we'll certainly make those plans public right away. We might well be better off if we take a little more time so what we present at DevCon has more solidity.

How did you choose the company name?

Soft as in software, and velocity as in speed of movement, in a specific direction.

You've had a lot of well wishes since taking over. What can the Clarion development community do to help get you off to a good start?

Buy everything we make![laughter] There are probably a lot of things customers can do. Open input is important to me. I'm a firm believer that people commit their time and their money to products and tools that they feel they have some input into, and I actively seek that. I think that is one of the most important ways to hear what should happen. There are probably two things related to that. One is that the newsgroups aren't really the place to put some of the comments that have been made that scare away potential users. You might be surprised at some of the emails I get from people who evaluate the product and go out and find the newsgroups. This is obviously prior to May 1, but there was a lot of negative stuff. And I'm not saying that there wasn't a reason behind it and there wasn't a lot of confusion, but it really tended to hurt us a lot more than people might realize. People acting responsibly in the newsgroups and using them as they were intended will have a bigger impact than might be obvious.

The other thing is we need people to move up to the current version. It puts an undue strain on our resources when we have to do a lot of support for versions that haven't shipped for several years. We still have people who call us and want support for 2.1

occasionally. It would really help us get off in the right start if we can show our investors and our developers and potential developers that our user community is behind us and with us on the current product.

Any closing comments?

The only thing that I would add is that we intend as a group to actively look for input from our users, and quickly deliver new technology as we see a need for it, and that will be evidenced by input from our users. And our philosophy on that is we want to make developers capable of using our technology as easily as they can run a wizard today. Simple is good. I believe you can measure technology not just by how powerful it is but also by how easy it is to understand and get the job done. And that's going to be part of our commitment. Everyone at SoftVelocity understands that we can learn a lot from our users, and we can do a lot more enabling them than has been done in the past, by means of support and documentation and education.

Reborn Free

CLARION *online*

published by CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5 by TopSpeed

FREE Microsoft Internet Explorer

etc2000 EVENT SPONSOR

The Novice's Corner: File Access Basics

by Dave Harms

Almost all Clarion applications work with data files of one description or another. Most use databases with numerous different tables and files, and perhaps even more than one database or file system. And all of this is ultimately possible because of Clarion's file access grammar.

From the start Clarion has had a set of simple, elegant functions for accessing data files. CDD introduced replaceable file drivers, which let developers switch from one file system to another without necessarily changing any code. Clarion for Windows added VIEWs, or logical files. Now in ABC there are a set of classes which wrap around the file access grammar and handle error checking and file opening/closing, and which for the most part obviate any need on the part of the developer to access the files directly.

Note that I said "for the most part." There are still times in ABC when you'll need to use a non-ABC file access function, and if you're a legacy code it's your only option. In this article I'll cover the basics of file access using just Clarion language statements.

File Drivers And Caveats

Part of Clarion's appeal for database development is its system of replaceable file drivers. You can use one set of statements for file access, and just by switching the driver work with a completely different system. An application that updates dBase files, for instance, could also be made to work with a MSSQL database by changing drivers. But not all drivers support the same data types or the same functionality, and often maximum performance means using features specific to one driver. Even within a product like Clarion, code portability exacts a price especially when using SQL (although ABC handles this much better than legacy code does). Look in the Help for each database driver for the heading Supported Commands and Attributes for driver-specific information on functionality and datatypes.

Creating Files

Should you need to create data files, just make sure you check the Enable File Creation checkbox in the file's dictionary properties. This will add the `CREATE` attribute to the file's definition. `CREATE` causes the compiler/linker to store information such as the field and key names in the application, rather than just basic symbolic information, which is more compact.

For flat file systems such as the TopSpeed file format, you can usually just run your application and any missing data files will be `CREATEd`. For SQL databases, you'll need to do this through database's administrative facilities, or using an SQL script, or perhaps the synchronizer (unless you're using ODBC, in which case the synchronizer isn't an option). Generally speaking when working with SQL databases it's better to import into Clarion than to try to use Clarion to create the database.

Opening And Closing Files

Before you can use a file (or table, if you prefer SQL syntax), you have to open it. In Clarion, this is done with the `OPEN` function:

```
OPEN(file, access mode)
```

The `file` parameter is the label of the file, not the name. When you define a file in your dictionary, the first field you fill in is called Name. This is a bit misleading. The dictionary name is actually the label and is what you'll refer to in your code when you want to do something with the file. If you don't specify a particular name for the data file itself, the label will be used. If you have a file called Names which uses the TopSpeed file driver, the default name of the file will be Names.TPS. But you can also specify a different name for the file. If you use a variable for the file name (in the dictionary, precede the variable name with ! in the Full Pathname prompt) you can specify the file name, including the path if you wish, at runtime. You might to this to create different datasets in different directories, for instance.

The second parameter to `OPEN` is the file's access mode. There are two parts to this: the current user's access, and others' access. Figure 1 shows the possible values.

Figure 1. File access modes.

User	Equate (from tplequ.clw)
Current User	ReadOnly EQUATE (0H) WriteOnly EQUATE (1H) ReadWrite EQUATE (2H)

Other Users	AnyAccess EQUATE (0H)
	DenyAll EQUATE (10H)
	DenyWrite EQUATE (20H)
	DenyRead EQUATE (30H)
	DenyNone EQUATE (40H)

The access mode is a combination of rights for the current user and other users. Typically a generated network (as opposed to single user) application will use a mode of 42h, **or** ReadWrite+DenyNone. **File operations which require exclusive access will typically use** 12h (ReadWrite+DenyAll) **or** 22h (ReadWrite+DenyWrite). **The latter allows others to at least read the data during the operation.**

You close a file with the CLOSE **function:**

CLOSE(*file*)

It's important to keep track of when a file is opened or closed. Consider a browse and form. The browse opens a file, and calls the form for updates. Perhaps you can assume that the form will always be called by this browse (a dangerous assumption). You have an OPEN(file) statement when the browse starts, and a CLOSE(file) statement when it ends. But what if that browse is called from another procedure that has a similar idea? The first procedure opens the file. The second (browse) procedure also opens the file, which has no effect, since the file is open. But when the browse ends it will close the file, causing problems for the first procedure which still thinks it has an open file.

In legacy applications code is generated which keeps track of a global counter for the file. Figure 1 shows example opening and closing code, with MyFile::Used as the global counter:

```
IF MyFile::Used = 0
  CheckOpen(MyFile,1)
END
MyFile::Used += 1

! procedure code

MyFile::Used -= 1
IF MyFile::Used = 0 THEN CLOSE(MyFile).
```

The CheckOpen function, which is generated into the *appname_SF*.CLW file, does all necessary error checking and will create files if necessary and permitted. As the code shows, the file is only closed when the counter hits zero, which should mean that there are no procedures still needing the use of the file. ABC uses a similar approach, although the code is contained in wrapper methods for the Open and Close functions. As you can imagine, issuing a Close on a file in embed code can have some unpleasant side effects. It's better to just let the templates handle file opening/closing.

> **NOTE:** If you're using the Source procedure template, you may have to write your own file opening/closing code as no such code is generated. It's

best to follow the standard practice for whichever template set you're using, ABC or legacy.

Accessing Data

Once you have a file or table in hand, you're ready to start reading data.

As Figure 1 shows, just a few statements handle pretty any kind of file access.

Figure 1. Core Clarion record retrieval statements

Function	Purpose
SET	Set processing order
NEXT	Read the next record
PREVIOUS	Read the previous record
GET	Get a specific record

The SET command, when used with files, has the following formats, as taken from the C5 help file (my notes in regular type and parentheses):

SET(file)	*Specifies physical record order processing and positions to the beginning (SET...NEXT) or end (SET...PREVIOUS) of the file.*
SET(file,key)	*Specifies physical record order processing and positions to the first record which contains values matching* (or nearest to) *the values in the component fields of the key. NOTE: This form is rarely used and is only useful if the file has been physically sorted in the key order. A common mistake is to use this form when SET(key,key) is the actual form desired.*(This usually happens when a GET(file,key) is changed to a SET, but which should be SET(key,key).
SET(file,filepointer)	*Specifies physical record order processing and positions to the filepointer record within the file.*
SET(key)	*Specifies keyed sequence processing and positions to the beginning (SET...NEXT) or end (SET...PREVIOUS) of the file in that sequence.*

`SET(key,key)`	*Specifies keyed sequence processing and positions to the first or last record which contains values matching* (or nearest to) *the values in the component fields of the key. Both key parameters must be the same.*
`SET(key,keypointer)`	*Specifies keyed sequence processing and positions to the keypointer record within the key.*
`SET(key,key,filepointer)`	*Specifies keyed sequence processing and positions to a record which contains values matching* (or nearest to) *the values in the component fields of the key at the exact record number specified by filepointer. Both key parameters must be the same.*

A few forms of SET cover almost all situations. SET(key,key) is probably the most common form, since it supplies the records in key order, beginning with a record matching (or nearest to) a specified value in that key. That's exactly what a browse needs. For processing all records in a file, in key order, you probably want to use `SET(key)`.

If you're going to be updating records and these updates will change key values, you may find that not all records are processed or some records are processed more than once. This is because you're changing the data used to determine the order of record retrieval. In such a situation, you have several options. You can use a key you're sure will not be affected; you can use `SET(file)` and process in record order without regard to keys; or (related to the first option) you can do a two-pass operation, store pointers to the affected records, and then retrieve each one individually via a unique ID on the second pass and update the values.

Retrieving Records With NEXT And PREVIOUS

It's important to remember that SET by itself doesn't retrieve any records; it simply prepares the file driver for a subsequent NEXT or PREVIOUS call. A situation encountered in countless browses is a user using a locator field.

The `SET` function works with whatever values are currently in the file buffer. In a list of names, if a user is searching for the last name of "Smith" in a file called `NAMES`, with a key on the `LastName` field, the file operations under the cover would go like this:

```
CLEAR(NAM:RECORD,-1)
NAM:LastName = 'SMITH'
SET(NAM:LastNameKey, NAM:LastNameKey)
LOOP
  NEXT(Names)
  ..
```

In reality this will be complicated a bit by some error checking code and the need to limit the retrieved records to the number of lines showing in the browse, but that's the general idea. The first time NEXT is called it will retrieve the first record matching

'SMITH', or the next record in the key if there is no 'SMITH'.

Note the use of the CLEAR statement before setting the key value. If you're using a single component key, you don't need to do a CLEAR (which is the example uses a -1 paremeter to set all fields to their lowest possible values). If you don't use CLEAR on a multi-component key, you may find that you don't get the records you expect to get.

A similar technique is used to accomplish autoincrementing of unique record IDs. In an autoincrementing key each value must be unique. In legacy apps the template-generated code to get looks something like this:

```
SET(myf:IDKey)
PREVIOUS(MyFile)
IF ERRORCODE()
  NewID = 1
ELSE
  NewID = myf:ID + 1
END
```

Here the SET(key) format is used to return the highest existing value in the key. If no record is found, it's the first and the value is 1. Otherwise the value is the highest existing value plus one. Again, the actual code is a bit more complicated, as it needs to preserve any primed fields, and also will try up to three times to get a new ID, just in case someone else adds a record between the time the current user gets the highest ID and adds a placeholder record.

In ABC the autoincrement code (hidden inside abfile.clw) uses the SET(key,key) format to support multicomponent autoincrementing keys:

```
CLEAR(AutoIncField,1)
SET(SELF.Keys.Key,SELF.Keys.Key)
IF SELF.Keys.Fields.Ascend
  PREVIOUS(SELF.File)
ELSE
  NEXT(SELF.File)
END
```

Here the CLEAR function sets the autoincrement field to its highest value via the 1 parameter (different data types will have different maximum values – let CLEAR sort it out) and PREVIOUS/NEXT are called based on whether the key is ascending or descending. The usual error checking and incrementing of the key value follows.

Coming Up…

In the next installment of the Novice's Corner I'll look at the GET function, and some of the intricacies of updating records.

[David Harms](#) is an independent software developer and the co-author with Ross Santos of Developing Clarion for Windows Applications, published by SAMS (1995). He is also the editor and publisher of Clarion Magazine.

Reborn Free

CLARION online

published by
CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5
by TopSpeed

FREE Microsoft
Internet
Explorer

etc2000
EVENT SPONSOR

LSZip 2.5 From Linder Software

Reviewed by Tom Hebenstreit, **Reviews Editor**

Zip files.

I got 'em. You got 'em. The web is full of them. Pretty much a day doesn't go by that I don't do *something* with a zip file.

Why? Because you can't beat them for grouping, compressing and storing other files.

Most of us use a standalone shareware or commercial application to do all the zip-related work for us. There are times, though, when it would be useful to be able to manipulate zip files directly from within a Clarion program.

And I mean *really* useful. For creating email attachments. For backing up data. For shrinking huge images or documents (such as Word or Excel files) down to a tiny sliver of their normal size so that you can store them directly in your data files. For combining multiple files into a single, easily distributable package. For…well, I'm sure you've gotten my drift by now.

So how do you, as a Clarion user, let your programs get into the act? Well, you *could* use `run()` or, for the API inclined, `ShellExecute()` to shell out to one of those other shareware programs, but then you can run into a world of troubles trying to make sure that the user has the program and version you are looking for in order for *your* program to do its job. Ugh.

If only you could just do it all yourself… (Bet you know what's coming next!)

Well, you can.

The subject of today's review is a product from Linder Software called LSZip, and it provides the keys to the kingdom when it comes to zip format files.

What actually is LSZip? It's a set of external libraries (DLLs) that you call directly from within your programs to provide full zip and unzip functionality.

You know, I have to confess that I am already a long-time user of LSZip, starting with

the first release about two years ago. I currently have multiple production systems using these libraries and have found them to be both stable and robust.

With the release of this new 2.50 version, though, it was time to put the product on the spot and give it the full Clarion Magazine review treatment.

Let's see how it fares.

Major Features

What's so great about LSZip?

In a word (ok, in four words): industry standard zip files.

How standard? Here's a quote from the LSZip documentation:

"LSZip is 100% compatible with PKZIP(R) 1.10, 2.04g and 2.x archives - even 'their' libraries don't have that. Most competitive products support 2.04g archives only."

Product highlights include:

- 16-bit and 32-bit language-independent DLLs. You can use LSZip with Clarion (2, 4, 5, 5.5), C++, VB and more. But unlike some other commercial compression libraries, LSZip was designed from the ground up to work with Clarion.
- Fast performance with a small footprint.
- Real-time message pump and callback support gives your programs the ability to display ongoing progress information and allows the user to cancel any operation.
- Support for in-memory compression (allows you to compress/expand information, images, etc., on the fly before storing it in your files).
- Support for operations such disk spanning (creating an archive that spans multiple disks). Includes the option to automatically remove all files from a disk before writing to it.
- Full support for storing paths and for folder recursion (zipping a folder and all sub-folders with it).
- Long filename and UNC support.
- Provides for comments on both the archive and individual file level.
- Customizable messages for internationalization.
- Can repair damaged zip files
- Automatically stores already compressed files without trying to compress them again.
- No runtime royalties
- And more…

All right, you get the idea. LSZip basically leaves no zip-related stone unturned.

Installation

To install the LSZip libraries, I downloaded the most current version from the Linder Software Web site (it weighed in at a bit under one megabyte). After clicking through a few introductory screens, viewing the license agreement and entering a password to unlock the install, I was presented with a list of all of the compatible versions of Clarion for Windows that I had installed on my machine (2, 4, 5 and 5.5, by the way). Yes, that

means that the one install package works for all versions of Clarion from 2.0 on up, and that it actually checked for installed versions so that I only saw appropriate choices. Nice touch, that.

Depending on the version I chose, the installer defaulted to the appropriate folder. For example, if I chose Clarion 4, it automatically supplied "E:\Clarion4". As I was planning on testing using C5.5 Beta 2, I selected that version and let LSZip install to my "E:\C55" folder. After it copied the files, it asked if it should register the templates. I said yes, it registered them without a hitch and then displayed a read-me file containing the final installation notes and instructions. Very smart and very smooth.

About the only knock that I have against the setup program it is that it does not install its files into the default Clarion \bin, \lib and \libsrc folders, but instead creates a new LSZIP sub-folder and install the files under there. This means that you either need to edit your Clarion Redirection (RED) file, or copy the files to the standard folders yourself before you can actually start using the libraries. The read-me file does contain instructions on editing the RED file, though, and you can just cut and paste from their example into your environment. I actually did both. I edited the RED so that the environment could find the LSZip files, and I copied the DLLs to my \C55\bin folder so that my test programs would be able to find them on the path.

A point to keep in mind is that the same LSZip DLLs are used by all versions of Clarion, and those names have not changed from prior versions LSZip. I got caught by that when I first tried running some of the example apps which used the newest features – I got an error message that the new function was not the library. Why? Because Windows had found and loaded an older version of the LSZip DLL in my Clarion5 folders (due to C5 being before C5.5 in my path string). This was easily rectified, but it is still something to watch out for. To help alleviate this, Linder also provides a handy standalone utility that can scan your drives and locate all instances of the various LSZip DLL files. The utility also displays the DLL version, which makes it easy to catch those outdated files.

One final note: the installation program, with its automatic Clarion version detection and template registering, was created using another Linder Software product called Setup Builder ([previewed](#) in the May 1999 issue). Judging from the quality and flexibility of this install package, the current release of Setup Builder looks worth checking out (hmmm… sounds like a future review to me).

Implementation

The difficulty factor for adding the zip and unzip libraries to your applications ranges from moderate to fairly easy, depending on whether you are using legacy or ABC templates. In both cases, though, be prepared to write some embed code. The only template supplied with LSZip is a global extension that takes care of the background details of adding the library global variables, prototypes and so forth to your application (one version is supplied for legacy, another for ABC).

Beyond that, ABC users get an additional boost from a set of classes that wrap around the libraries and simplify their use. Legacy users (or ABC users, if they wish to do so) call the actual library functions directly.

Neither situation is particularly difficult. The LSZip library functions are well documented and are no harder to use than any other Clarion function call.

Figure 1. The LSZip global extension (same options for legacy and ABC).

```
 LSZip Compression Library Varaible Definitions

 Copyright © 2000 Linder Software
 www.lindersoftware.com  - support@lindersoftware.com
 All Rights Reserved Worldwide

 Release: 2.50.0306

 LSZip Library Support
 ☑ Compression (ZIP)
 ☑ Decompression (UNZIP)


 LSZip Services
 ☐ Use LSZip Function Library
 ☐ Use LSZip Error Messages
```
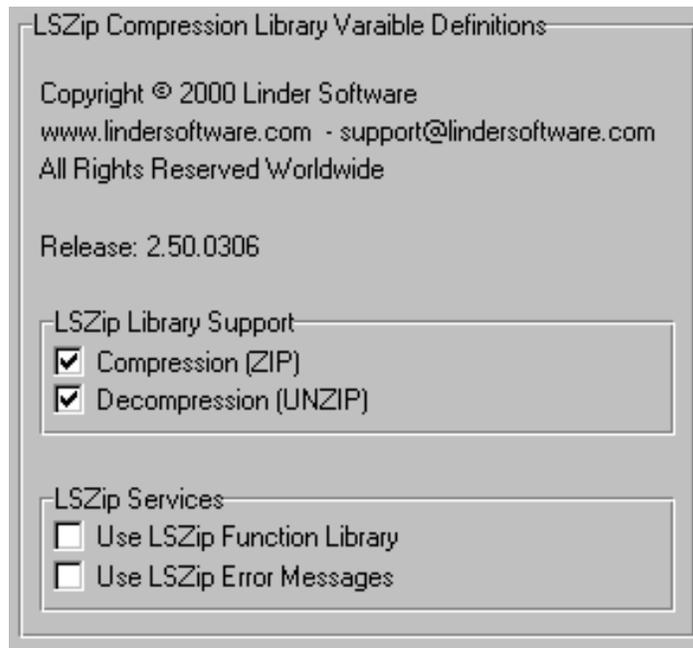
To illustrate the differences between direct use and calling the ABC compatible classes, here are some code snippets that perform the same basic task of creating a zip archive and then adding a group of files. By the way, the majority of code shown here comes from one of the many example programs that come with LSZip, modified by me to add a few more functions.

The Direct Method

To call the LSZip library functions directly, begin by registering the callback functions that lets you display progress bars, etc. for the user.

In the EVENT:OpenWindow embed is the following:

```
!***INIT LIB MESSAGES ***
!'Update' Progress Bar
W_MESSAGE    = 'LSZP'
LSZ_PROGRESS = RegisterWindowMessage( W_MESSAGE )

!'Abort' operation
W_MESSAGE = 'LSZA'
LSZ_ABORT = RegisterWindowMessage( W_MESSAGE )

!Get item info #1
W_MESSAGE  = 'LSZI'
LSZ_INFO_1 = RegisterWindowMessage( W_MESSAGE )

!Install Progress Bar Message Pump
SavedProc1 = ProgressWindow{PROP:WndProc}
ProgressWindow{PROP:WndProc} = ADDRESS( MP:Progress )

!Install Real Time Abort
! Save this procedure
SavedProc2 = ProgressWindow{PROP:ClientWndProc}
! Name subclass procedure
ProgressWindow{PROP:ClientWndProc} = ADDRESS( MP:Abort )
```

```
!Initialize
FirstWindowHandle = ProgressWindow{PROP:handle}
IF LSZIP_INIT( FirstWindowHandle ) <> 1 THEN
   MESSAGE('Could not initialize zip message pump')
END
```

Now, before you run screaming for the exits, I must point out that *all* of the above code deals with setting up the callback functions. If I didn't want to use them in this case, none of the above is required.

Moving on, here is the basic code needed to tell the library what needs to be done, and then to tell it to actually do the work. Note that in the interest of condensing the listing, I eliminated some error checking that could otherwise be done.

In the EVENT:Accepted embed of a 'Go' button:

```
! Initialize registration
! Your user name
UserName      = 'Unregistered User'
! Your user PIN
UserPIN       = 1381676767
! Register library
IF LSZIP_PIN( UserName, UserPin) = 0 THEN
   MESSAGE('You are not authorized to use this library')
   HALT
END

! Specify archive file name, file spec, etc.
! ZIP archive name
szArchiveName = 'BACKUP.ZIP'
! File specification
FileSpec      = szTargetPath & '*.*'
! ADD method
zLibMethod    = zm_ADD
! FASTEST compression
zCompLevel    = 1
! No library flag
zFlag         = ZIP_NO_FLAG

! Pass filespec to the library
z# = LSZIP_INCLUDE( FileSpec )

z# = LSZIP_EXECUTE(szArchiveName,zLibMethod,|
  zCompLevel,zFlag )

IF (z# < 0) THEN
   MESSAGE('ERROR #: ' & RetValue)
END
```

That's not so bad, is it?

If you look at the above code, you'll actually see that it only took two function calls to

create a zip and add the files to it. The other function (`LSZIP_PIN`) tells the library that yes, you are a registered user. If that function is not called, or invalid registration information is used, the libraries will pop up "Unregistered Version" messages every time they are called.

Using The LSZip Class In An ABC Application

Ok, now the same task under ABC using the LSZip classes. First, you need to declare the class objects, like this:

```
Zip         ZipClassType
ZipFilesQ   ZipResultType
```

Next, the Zip object is initialized in the `EVENT:OpenWindow` embed:

```
Zip.Init( Window, ?Progress1, ?Progress2, |
  ?String1, ?List1, ZipFilesQ, LIC:UserName, LIC:UserPin )
```

Whew! Looks like a lot of parameters, but the only two that are really needed are the last two (they register the library). The rest of the parameters take care of setting up the callback functions so that the user will see a progress bar for the current file being processed (`?Progress1`), the name of that file (`?String1`), a progress bar for overall progress (`?Progress2`), and a list box whose queue is automatically filled with each completed file (`?List1` and the aforementioned `ZipFileQ`).

And yes, that one line does more than the dozen plus lines shown above in the direct code example.

Ok, now for the nitty-gritty code to do the work. Note that I also tossed in a couple of new lines to set the compression level and to turn on folder recursion depending on user input.

```
! Define the zip archive name
Zip.SetArchiveName( TargetArchive )

! Define all files to be zipped up
Zip.DefineFileSpec( BackupPath )
Zip.AddFileSpec( '*.*' )

! Set Compression Level
z# = Zip.SetCompressionLevel( UserCompLevel )

! Add flags to control behavior
Zip.AddFlag( ZIP_QUIET_FLAG )
Zip.AddFlag( ZIP_AUTOMATICSPAN_FLAG )

!!TH!! Recurse and store paths if user checked box
If UserRecurseFolders
    Zip.AddFlag( ZIP_Recurse_Flag )
    Zip.AddFlag( ZIP_StorePath_Flag )
End

! Execute zip command
Res = Zip.LSZExecute( zm_ADD )
IF Res THEN
    MESSAGE('Error: ' & Res)
```

```
END
```

In this case, the primary benefit of the class is not fewer lines of code, but more readable lines.

What does all of this look like to the user? Here is the window from the LSZip example program, along with my additions to set the compression level and turn folder recursion on and off. I captured this image in the middle of a run so that you could see the various progress bars under the processed files list box (current file, file name and overall progress, respectively.)

Figure 2. LSZip in action.
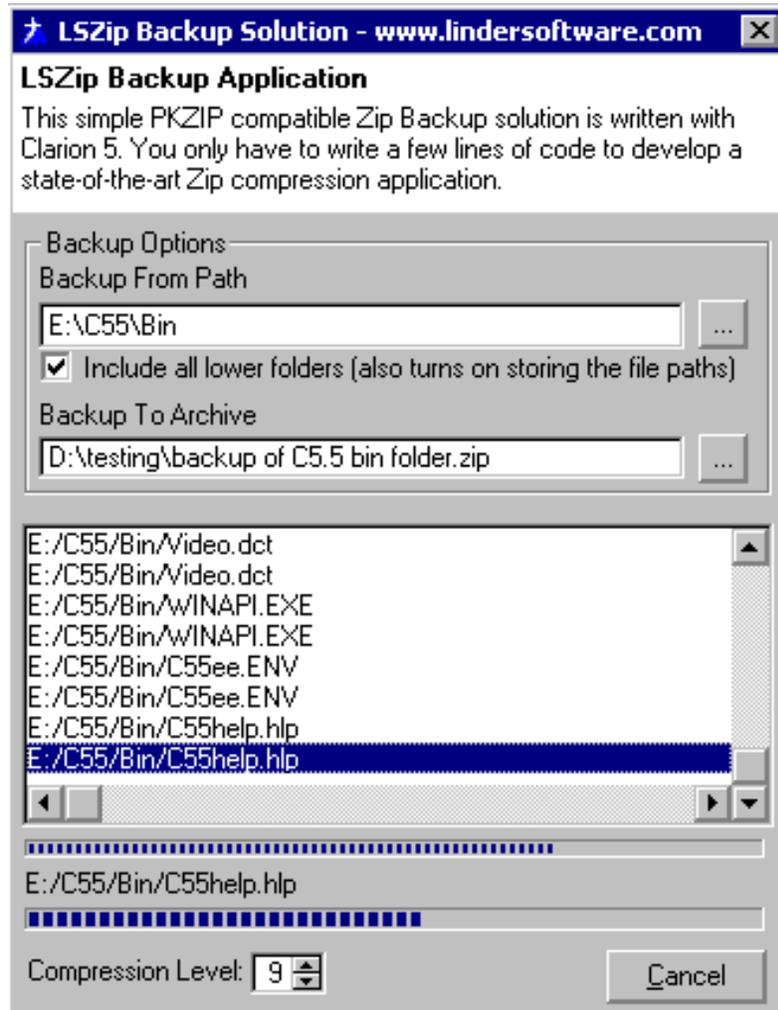
Keep in mind that all of those progress updates (including filling and displaying the list of processed files) are being done automatically by LSZip, using just those lines of code listed above. All I had to do is call the zip library. Sweet.

By the way, if I was using ABC and wanted to cheat a bit for this particular task, I could create an archive and fill it with one line:

```
Zip.QuickZip('c:\mydata.zip','c:\mydir\data')
```

This shortcut method will create a zip archive (the first parameter) and then add all of the files in the specified folder to it (the second parameter).

All in all, with these short examples you can see that while implementing LSZip isn't a

no-brainer, it's not rocket science either. I'd love to see more templates in the product to make it more accessible to the less code-inclined, but the LSZip classes for ABC apps are a definite step in the right direction.

Performance

Performance and zip compatibility in all cases was excellent.

I tried many combinations using WinZip 8.0 and LSZip, creating various types of archive with one tool and then checking/processing it with the other to see if I could find any problems. I am pleased to report that I failed in every attempt to trip LSZip up.

I also did a few rough speed tests where I zipped my entire C5.5 \bin folder using maximal compression and folder recursion. I found LSZip and WinZip to be within a few seconds of each other at compressing close to 40 Megs worth of files down to about 18 Megs. If there was a winner, I say LSZip was perhaps a few seconds faster. How long did it take? About 1 minute, 15 seconds using a 500mHz Pentium III.

Documentation

With the release of version LSZip 2.5, Linder has re-written the documentation and switched to providing it in the form of an Adobe Acrobat PDF file. I found the file to be thorough and well formatted, but still a bit rough around the edges.

Probably the biggest problem with the current format is that it is very difficult to jump around within the PDF file. For example, when looking at an LSZip class method, I often found it useful to check out the documentation for the underlying function it was calling. I found myself having to constantly go back to the table of contents, find a page number, jump to it, and then do the same in reverse to get back to where I was when I started.

In a nutshell, the PDF is sorely in need of a navigation pane where you can just click on a table of contents item and go directly there. My final workaround was to print just the table of contents and keep it on my desk for reference.

In case you're curious, the PDF is 213 pages long but it's not as scary as it sounds. For clarity, in many places it uses a "one function definition per page" format, but that also means there is a lot of air in there as well.

LSZip also includes over a dozen example programs that illustrate specific features. The examples are split between legacy and ABC, which is handy for comparing the two approaches (direct calls using legacy and ABC using the LSZip class).

Technical Support

Support is provided direct from Linder Software via email.

Linder Software's support has been uniformly excellent in all the time I have been using LSZip (and other Linder products). Questions are answered completely, and I have found them to be very receptive to suggestions for new features.

Summary

As I said at the beginning of this review, I have been a long-time user of LSZip, and have found it to be invaluable in situations ranging from simple data backup to

automated distribution of files via the Internet. With the addition of the new in-memory compression routines, an already excellent product has become even better.

Is it perfect? No, but careful readers will have noticed that most of my gripes and suggestions have had to do with ancillary items such as ease of use and documentation, not with performance or reliability (both of which are stellar).

Bottom line: If you have any type of archival or compression needs, I can't recommend LSZip highly enough. Check it out!

PRODUCT RATING	
Overall	▲▲▲◢
Ability to do the task	Excellent
Ease of use	Legacy - Good ABC - Very Good
Ease of installation	Very Good
Documentation	Good
Technical support	Excellent
Black box DLLs/LIBs	Yes

LEGEND	
First class all the way	▲▲▲▲
More than adequate	▲▲▲
Barely adequate	▲▲
Don't even think about it	▲

LSZip is $179.00 USD, and can be purchased directly from Linder Software or via secure online purchase.

For more information, visit their web site at: http://www.lindersoftware.com

Vendor Comments from Friedrich Linder of Linder Software:

We are committed to continuing to enhance the LSZip product, and all points and suggestions that Tom raised during his review will be seriously considered. Documentation will be increased and improved and more templates will be added to make it more accessible to the less code-inclined.

You can do so much more with LSZip than just backing up your data. The new 'in-memory' routine makes it possible to compress images and large text directly into BLOBs and to decompress on the fly from BLOBs into controls. The result is a massive saving in disk space and disk or network access.

Reborn Free

CLARION *online*

published by
CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5 *by TopSpeed*

FREE Microsoft Internet Explorer

etc 2000 EVENT SPONSOR

etc Reception And Day One Highlights

by Dave Harms

It's not over yet, and I don't want to jinx anything, but when Lee White puts on a conference, it's going to be done in style. So far, so good!

This year's East Tennessee Clarion Conference got underway yesterday afternoon with an excellent presentation by James Fortune on application help, and how to avoid writing it. It was a bit of a rough start for James as he'd come across the pond with everything on a zip disk, and it took longer than expected to get the supplied NT box up and running (little mixup with the password, I understand). With no time to get the audio working, James did an admirable job singing and dancing his way through the missing effects. Formerly more focused on writing documentation, he's now directed his efforts at good application design which obviates the need for most help. Look for more on James' presentation later in Clarion Magazine.

Following this presentation the attendees gathered for a reception, meaning a whole lotta Mexican food, appetizers (served after the meal so they wouldn't be mistaken *for* the meal this time) and the sweet strains of a string quartet accompanying the chinwagging (mpeg, 336K). Many of the usual suspects were in attendance, with plenty of new faces as well to put the total, including guests, at over 150. The number of developers in attendance is up over two years ago, not surprising given the good press the original etc garnered.

The weather was fairly good Tuesday afternoon, but it's thunderstorm season in this part of the country and by the time the reception wound down and the attendees headed to their rooms (or their beverage rooms) there'd been a slight turn for the worse. Around 4 a.m. the storm really ramped up, with thunder reverberating down the valley and lightning jagging though the clouds.

As in 1998, the conference is at Gatlinburg's Edgewater Hotel, which as its name suggests sits right alongside a pleasant mountain stream. Until the heavy rains hit, that is. Before the first Wednesday session the civil defense siren sounded, warning of flash flooding in low-lying areas. The water had risen as far as the bridge between the hotel and the road (a little higher than the pictures show), but hotel staff indicated that the water had been over that bridge many times before, so not to worry. The intermittent sirens ended before the first session.

Tuesday Afternoon   **Wednesday Morning**

Bob Zaunere, SoftVelocity's President and CEO, was at etc for the day and spent a half hour at the start of the session addressing the attendees and taking questions. I'll have the details of that session shortly. Bob had a receptive audience, and tipped his hand about some new Clarion features on the way. For a guy who's no doubt been run off his feet with new duties, he appeared relaxed and seemed to be truly enjoying his job, the conference, and meeting with developers.

After Bob's talk there was a short break, and then some guy from Clarion Magazine prattled endlessly about web site development, and thankfully only one attendee needed attention from paramedics. Rumor is there will be a report on that presentation sometime soon.

Fortified by lunch, we headed back into the lecture hall for Nik Johnson's wide-ranging presentation on Clarion development tips and tricks, with emphasis on classes and templates. Nik's probably forgotten more than most of us will ever know about what the Clarion development tools are really capable of, and how they can be used effectively. Details to follow.

After Nik wrapped things up Lee & Co. held the door prize raffle. With almost one giveaway for every two developers, there were a lot of happy campers, but the star attraction was (cue the reverb):

RAFFLEMAN 2000

Coming soon to a store near you ([mpeg, 1080K](#)). Lee's totally awesome Raffleman 2000 filled the screen, randomly selecting numbers to match those on attendees badges. Each winner had the choice of whatever prizes remained, and some of these prizes weren't exactly cheap.

For supper we were all supposed to pile into trolleys and head out to the park for a Cajun cookout. But when the cooking team got up early this morning (or were awakened by early morning lightning and thunder) it was pretty obvious that the weather wasn't going to cooperate. The feast has been postponed until Thursday night, at which time it *will* go ahead, no matter what the weather.

Aside from the Cajun cookout, Thursday's schedule includes Steve Parker on Web Builder in the morning, and Skip Williams explaining non-Web Edition internet development in the afternoon. On Friday Andy Stapleton will look at still other web development options, and Saturday Bruce Johnson is teaching a full day class on ABC and embed code.

The etc format is a bit different from that of most other DevCons. There are no track sessions – instead, all attendees see the same presentation, and each speaker is given three hours, except for Andy, who sneers at fiddling short half-day sessions. This conference format makes for much more detailed information than is usually available, and removes the old problem of having to decide among your three favorite sessions, all of which are scheduled for exactly the same time.

More to come - stay tuned!

ETC Starts Off In Style
(May 24, 2000)

Bob Zaunere Keynote Address
(May 24, 2000)

TopSpeed SA Posts C5.5 Press Release
(May 25, 2000)

TopSpeed SA Posts C5.5 Press Release
(May 25, 2000)

Clarion MAGAZINE

Bob Zaunere: etc Q&A

Wednesday morning Bob (Robert) Zaunere, SoftVelocity's
President and CEO addressed the etc conference. As is
apparently becoming the norm, the Team TopSpeed (Team
Velocity?) contingent did "the wave" as he took the podium.
Zaunere began by saying one of the most important things he
thought he could do was open up communication with the
developers. He read from some notes he'd prepared to
answer questions that had arisen since his Clarion Magazine
interview, and finally took questions from the audience.

A lot of people have asked about SoftVelocity's
development plans. Zaunere defined the company's
immediate mission as finishing Clarion 5.5 and making it a very solid release. Nigel
Hicks is now in charge of the beta builds, and was described by Zaunere as a very
pragmatic guy who checks code thoroughly before shipping it out. It will probably at
least another few weeks before a final release is ready. Following 5.5, SoftVelocity plans
to provide a number of incremental releases which will have new technology focused on
standards such as COM, DCOM, XML/XSSL and other product improvements.

SoftVelocity will use developers who have expertise in whatever technology the
company wants to deliver, and will form alliances and integrate with other companies'
software. SoftVelocity will also be working much more closely with third party
developers.

Zaunere spoke of the company's renewed energy and purpose, as evidenced in the new
feature list. Clarion 5.5 Professional and Enterprise Edition will both ship with the
following (except as noted):

- Developer copy of the TS ODBC driver
- Integrated RTF support with a template wrapper using a Clarion DLL
- Start of a collection of code templates wrapping the most-used Windows API calls
- PE will ship with the MS SQL driver (already in EE)
- EE will include a single-server license for the Oracle driver
- Integrated support for HTML help at runtime

The above represents effort of the last three weeks.

Zaunere then spoke of SoftVelocity's marketing plan. The company has been
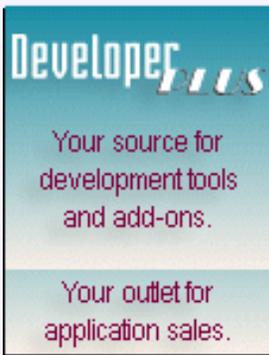
approached in the past four to six weeks by universities interested in teaching Clarion, and this will be pursued. An entry level edition is also under consideration; this would be bundled with a CBT course focused on getting started with Clarion. SoftVelocity is also looking at conducting free web-based seminars, followed by short web-based training seminars. Corporate IT departments are also a target; the strategy involves finding areas where Clarion can be useful in a project with little initial investment. Zaunere will be canvassing developers for more input in the near future.

A number of issues came up during the question and answer period, with Zaunere's responses summarized as follows:

- The old policy of not allowing Clarion to be used in courses unless the Clarion name appeared in the course literature is just that – old policy.

- Printed manuals will be available for 5.5 on request, but the product will ship with PDFs.

- The large number of embed points in 5.5 is a problem; only a dozen or so are commonly used, and the embed tree could be altered to display these by default, or the environment could do some learning about which embeds are most-used (this is technology used in the wizatrons).

- A lot of work has been done on the 32 bit IDE, but it's way too late in the development cycle to introduce it in 5.5. The next major release of the product, which will *not* be 12-18 months away, will be totally 32 bit. Developers wanting to do 16 bit development will need to use a previous release.

- SoftVelocity's relationship with Sensium is that both came from the same parent company. SoftVelocity's relationship with the Development Centre is, as before, one of contracting for development services. This is a non-exclusive contract; SoftVelocity can use other development resources, and the DC will have other demands on its time so will not exclusively do development for SoftVelocity.

- Patches need to be released as needed, i.e. more promptly than in the past.

- There's no reason why a bug database can't be made available to registered users through some suitable mechanism.

- SoftVelocity didn't take the name TopSpeed because TopSpeed Corporation needed it for a while, but SoftVelocity does have rights to use the TopSpeed trademarks.

- Wizatrons haven't had much support from users, and while the theory is great and the technology is sophisticated, they haven't yet proven viable.

- The application broker has a good future. SoftVelocity is looking at ways to improve it and make it more scalable and reliable.

- In the past Clarion has often been marketed as a replacement product, which suggests that the developer using the other product isn't as smart as he or she might be. It's a lot easier to sell Clarion as something that integrates with other tools.

- TS originally worked with Data Junction on the Clarion file format, and began collaboration on the TPS format. The Data Junction people decided to go with the ODBC driver and ran into a snag, which is fixed with the next release of 5.5.

- The subject of generating C code has come up before, and Zaunere has had conversations with people in recent weeks about generating code for other languages. He's definitely in support of any such activities.

ETC Starts Off In Style
(May 24,2000)

[Bob Zaunere Keynote Address](#)
(May 24,2000)

[TopSpeed SA Posts C5.5 Press Release](#)
(May 25,2000)

Clarion 5.5
New Features and Database Support

In response to changes in the marketplace and the requests from our community, SoftVelocity announces additional database support and a set of new features to be included in Clarion 5.5. Combined with the Visual Property Editor, the new Web Technology and the iBuild technology currently found in C5.5 beta 2 these changes make Clarion 5.5 the most versatile development tool we have ever offered.

New Product Features Included in both Professional (PE) and Enterprise Editions (EE)

Rich Text Support:

Add Wordpad style capability to your Clarion applications. Provided as a Clarion DLL with a template wrapper, does not require any external programs or ActiveX controls. (32-bit only)

- Supports search and replace, URL recognition and highlighting, file open, file save, copy and paste to and from the clipboard, supports windows fonts including colored text, bullets, numbering, text alignment and more.

- Can be used with Clarion memo fields.

- RTF 3.0 specification.

HTML Help Class and Template:

Add support for the new standard Help system for the Windows platform. HTML Help offers some distinct advantages over standard HTML, such as the ability to implement a combined table of contents and index and the use of keywords for advanced hyperlinking capability.

Crystal Reports Interface (continued):

- Provided as a DLL and Class interface with a template that provides instant access to the power of Crystal Reports 8.

- The Class interface makes it easy to add Crystal Reports functionality to a Clarion 5.5 application. The Crystal Report runs inside a Clarion window, making integration with your application seamless.

- Includes support for drilldown, report export, printer selection dialogs, summary view and more.

Utility Library:

- The start of a collection of often-used Window API calls complete with Code template wrappers for easy implementation. The library provides methods to read/write to the Windows registry, validate OLE objects, check the OS version, get the user name, minimize your app to system tray and many more functions.

Additional Database Support in PE and EE:

- **The Microsoft (MS) SQL Accelerator:**

TopSpeed ODBC Driver

Oracle Accelerator

MS SQL Accelerator

Rich Text Support

HTML Help Support

(32-bit only)

- Provided as a DLL with Extension templates for both Legacy and ABC as well as Code templates for the Class interface that communicates with the Microsoft HTML system.

- Easy to add support to either Generated Applications or hand-coded programs.

- Intercepts Clarion Help calls, and instead of calling the Windows Help system, it re-directs these calls to the Windows HTML Help system.

- Developers can use HTML Help to create on-line help for their application or create content for a multimedia title.

Crystal Reports Interface:

Add additional ease-of-use, flexibility and powerful reporting technology to your Clarion applications by incorporating support for Seagate Crystal Reports 8.

Provides native access to MS SQL Server databases.

- **TopSpeed ODBC Driver Developers License:** Allows you to provide built-in ODBC access to the TopSpeed database. This version supports Access 2000 and Crystal Reports 8. The license is for in-house development and evaluation, additional user licenses are required for deployment.

Additional Database Support in Enterprise Edition

- **Oracle Accelerator** (single server license): A 32-bit direct driver for Oracle, the Oracle Accelerator provides transparent access to the Oracle database for Clarion developers.

Crystal Reports Interface

Utility Library

SoftVelocity, Inc.
150 E. Sample Road
Pompano Beach, FL 33064
(954) 785-4555
Fax: (954) 946-1650
e-mail: sales@topspeed.com
www.topspeed.com

Reborn Free

CLARION online

published by
CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion 5 by TopSpeed

FREE Microsoft Internet Explorer

etc 2000 EVENT SPONSOR

Clarion News

May 31, 2000

Fomin Report Builder Help File

With the aid of Oleg Fomin, Scott Daughtry has created a comprehensive help file for Fomin Report Builder targeted at entry-level computer users. This help file uses over 80 screen captures and various tutorials to assist users in developing reports for various needs. The help file will continue to develop as time permits; feedback is encouraged.

LOGICBit Voodoo Discussion Board

LOGICBit has opened a web-based discussion board related to programming in general or programming with Voodoo. Subjects do not need be Voodoo related but please use a descriptive subject line. If you're having a problem, simply cut and past a code snippet along with your question. The purpose of this board is primarily for support. Secondly it allows users of the Voodoo library to post any questions they may have to several Voodoo users at once and to allow other Voodoo users to answer questions as well.

Free Voodoo Software For User Groups

LOGICBit will give away one (1) copy of Voodoo Shaman (full version) as a door prize for your next usergroup meeting. To participate, you must be the leader/president of your user group. Send your name, the date and time of the user group meeting, and the name of the user group to LOGICBit@aol.com with the subject line "Usergroup". For your usergroup to qualify, the group must have been in existence for at least 1 year. You can elect how you want to give the prize away but the prize must be awarded at the meeting to a participant (physical not virtual).

Makeover 2.0 Shipping

CapeSoft has released a number of product updates. Makeover 2.0 is now shipping, with a slight increase in price to $59 due to take effect at the end of June (the upgrade is free to owners of Makeover 1.0). Curent price is $49. Version 2 includes 14 new styles, over 60 new buttons, and 15 toolbars. The Style editor (Styler) has been significantly improved with lots of new features like Multi-Level Undo, Cut & Paste, Ship Lists, and so on.

WinEvent 2.6 Released

WinEvent is one of the lesser known of the CapeSoft products. It bundles together a

---

Clarion COM: As Easy As Visual Basic COM
(May 9, 2000)

Clarion Challenge: COM Examples
(May 9, 2000)

Cloning List Boxes - Source Update
(May 9, 2000)

Read The April 2000 News

numebr of functions for the more advanced programmer. Included are RS 232 Communication functions, Taskbar functions, native Windows Message trapping and more. With the release new modem functions have been added, and the ChangeIcon template has been added. $30.

Special Agent 1.22

A minor update to Special Agent (recently reviewed in Clarion Magazine) is now available. Free as always to registered users. Microsoft have upgraded all their Agent, Character and Speech files to be compatible with Windows 2000. So if you're distributing to W2K users than make sure you have the latest files from either www.microsoft.com/msagent or www.capesoft.com.

File Manager 2.91

File Manager release 2.91 consists of a number of internal bug fixes, and optimizations.

TearOff 1.55

This update fixes one or two outstanding bugs.

ClarionNET Beta 2 Available For Download

ClarionNET Beta 2 is now available for download. The Developers Demonstration package contains the complete ClarioNET development system. It includes all software developed to data in Beta 2 form: server and client libraries, extension template, documentation, and samples. You can use this Developers Demo to web-enable Clarion applications of your own. It is fully functional except that the client will always display a "Unlicensed Demonstration Version" window when launching the server program and at regular intervals during the work session.

Win32 Controls Special Offer

New from solid.software is the SysPack bundle, containing SysAni, SysTrack and SysHotkey. All three wrapper classes for win32 controls are available for 69,- EURO (that's 18,- EURO off). Offer limited until June 15, 2000.

BevelTool Freeware Available

Source code for BevelTool, a utility to create the value for PROP:BevelStyle, is now available.

ARCO Word Reporter New Release

A new release of ARCO Word Reporter is now available. This product lets you use MS Word as your report and printing engine for any kind of document. ARCO Word Reporter lets you and your customers create and change the layout of the documents in MS Word itself, just like any other template.ARCO Word Reporter is compatible with Clarion 4 - Clarion 5.5. Both ABC and Legacy templates are supported. New features include passing RTF formatted text, passing graphic files, merging created documents into one document (i.e. for mail merge), and the ability to call Word macros. Through June 12th, 2000 get 15% off new licenses, 20% off updates.

Dalby Source Printer 5.0 Price Drop

DSP 5.0 is now available for $29. Includes fixes to reported problems with Clarion 5, support for long filenames, a "print only modules that contain embeds" mode, support for include files, and removal of the 12 open files limit.

Sites Using C55 Web Technology

Antonio Miranda has a page listing publicly accessible sites which use Clarion 5.5 web technology.

SysTrack 1.1 Released

SysTrack from solid.software is a wrapper class for the Win32 trackbar/slider control. Version 1.1 fixes several bugs. This update is free for all registered users.

May 16, 2000

PD One Touch Date Tools B3b

Phil Will has uploaded a minor fix to PD One Touch Date Tools Beta3 which fixes compile errors for multi-DLL and legacy development, and a Windows 2000 demo GPF. B3b also adds schedule handling for ranges of dates with recurring and multi-day records. A demo is available.

Clarion User Interface Site

Steffen Rasmussen has set up a web site to collect as many examples of different user interfaces as possible to inspire Clarion developers. Includes examples of an Outlook-style menu.

SearchFlash 1.2 Released

SearchFlash 1.2 has been released. New features include multi-language support (English, German, Spanish, Dutch, Danish and Norwegian), and BEGINS WITH now checks the full length of the search string, not just the first letter. All source code is supplied.

Gitano Software Utilities Updated

The following utilities have been updated and can now be downloaded: G-RegPlus has a fix to a copy problem in the tech support form, as well as a bug fix to the calculator DLL. G-Calc fixes a calculator GPF, G-Buddy has new color schemes and a stray message screen has been removed.

New Product From solid.software

New from solid.software is SysHotKey, which wraps around the Win32 hotkey control (the one used for assigning keyboard combinations to desktop shortcuts), making it easy to use in Clarion while providing access to all the options the API provides. Demo version available for download. The full version is 29,- EURO.

May 9, 2000

New Clarion Boss To Attend etc

Bob Zaunere, president and CEO of SoftVelocity Inc, will be making an appearance at the East Tennessee Clarion Conference in Gatlinburg later this month. SoftVelocity, which was spun off from TopSpeed Corporation in early May, owns the rights to the Clarion product line. Zaunere will be at the conference to meet with Clarion developers and answer questions about SoftVelocity's plans for Clarion.

MessageEx Update Available

A new version of MessageEx is now available. This update includes several minor bugfixes, support for compiling apps with the local runtime library, and C5.5 B2 support. This update is free for all registered users.

Buggy 1.01 Update

An update to Buggy is available to all registered users. An updated trial version will also be available.

G-Cal Pro Beta

Gitano Software's long awaited scheduler is now in the beta testing stage! For a limited time you can join the BETA program and save $100.00 off the purchase price. The BETA program ends May 15th.

SysAni Update

A new version of SysAni is now available. New features include playing linked-in animations, improved ABC templates, added event handlers for START and STOP notifications, improved documentation, and support for Clarion 5.5 Beta 2. This update is free for all registered users.

PD One Touch Date Tools Beta 3

ProDomus has released Beta 2 of its "One Touch" Date Tools. This includes a new scheduling class with related templates. This release completes the planned feature set of the new Schedule Class. As with other ProDomus tools, it is internationalized for access to the global economy. Date Tools is available for both Clarion 5 and 5.5 Beta 2.

Voodoo Credit Card Purchases Available

Logicbit is now using DeveloperPLUS for online purchases. Introductory prices are $54 USD for Voodo, and $74 USD for Shaman. Voodoo and MMlib32 users can purchase an upgrade to Voodoo Shaman online for $24 USD. Prices good through May 15th. Voodoo is a low-level imaging library that lets you create visually appealing graphically intensive applications, interactive multimedia publications and games natively in Clarion for Windows. You can mix text, pictures, animations, video, sound and music in real-time. You can also retrieve images from scanners and digital cameras natively from your applications. No special hardware or software required. The core routines are written in tight 32-bit assembly and C/C++ to enhance and extend Clarion 4, 5 and 5.5 ABC or Legacy.

New Product: SysTrack

New from solid.software is SysTrack, the second in the company's product series of wrapper classes for common controls of the win32 api. SysTrack wraps around the trackbar control (also often called a "slider"), making it easy to use in Clarion while still providing access to all the options the API provides. The full version sells for 29,- EURO.

Open Source PIM Structure

The OpenDB Alliance is an association of Database creators and vendors committed to promoting a Open Source model of a Database structure as an industry-standard for the utilization and exchange of Database information and easier production of new applications. The Alliance is looking for people to duplicate the working example of its "Public PIM" in as many different varieties of software as possible including web pages or applications.

Reborn Free

CLARION online

published by
CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5
by TopSpeed

FREE Microsoft
Internet
Explorer

etc2000
EVENT SPONSOR

Bear Adds Variety To Cajun Cookout

by Dave Harms

Bear wasn't on the menu for Thursday's Cajun Cookout, but that didn't stop a yearling black bear from stopping by the cookout's secluded mountain setting several times, causing a certain amount of excitement and a little alarm.

Although no one was hurt by the bear, Larry Teames, assisting Cajun master chef Andrew Guidroz, suffered a serious flesh wound, reportedly while making some very sharp rice (if you look closely you can actually see some blood). Mortal kitchen injuries and rampaging fauna aside, the evening was another triumph of Cajun cookery and skillful beer delivery.

There was also some skillful bear delivery to come as a trap set by park rangers after the cookout nabbed a hungry, furry critter weighing in at about 250 lbs.
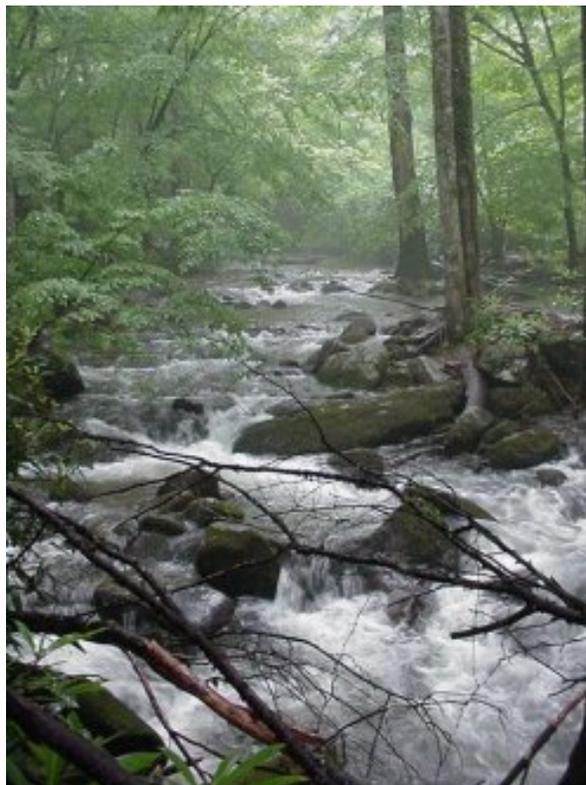
As in 1998, the feast took place at a cooking shelter in the Great Smoky Mountains, near one of those idyllic mountain streams that had, the day before, no doubt been a rushing torrent.

The original plan was to have the cookout Wednesday, not Thursday. At 5:00 a.m. the thunder and lightning was almost continuous, the rain pounding. Lee White woke (if in fact he slept at all), looked out (as if he needed to), said, "Postponed." Lee and Andrew headed to the kitchen, told the pig it had to wait another day, and then the cooking team went for refreshments.

As it turned out, Wednesday evening wasn't that bad, but the crew needed all day to prepare the meal, and it wouldn't have been any help to be surrounded by a torrential downpour. And the rains were torrential, causing localized flooding in Gatlinburg.

Thursday morning the team members were up early again. I don't have any pictures of the daytime preparations (yet) but I can guarantee there was a whole lot of cutting and chopping and cooking going on. Oh yeah, the pig had to be prepared.

The Cajun microwave
(MPEG, 256K)

The pig
(MPEG, 272K)

Trolley greeting
[low light]
(MPEG, 1040K)

Cooking a whole pig is an all-day kind of thing. And this wasn't just any pig, but a pig in a coffin. That's a steel-lined wooden box, which is overlaid with a coal tray. The pig was injected with secret herbs and spices. Isn't this how Colonel Sanders started out?

Attendees started out with appies, including pork sausage and boudin, which is ground pork and liver along with cooked rice, spices, and vegetables, stuffed into a sausage casing and then steamed. Fantastic stuff!

Carl Barnes (left) had a hand (so to speak) in serving the meal, as did Sue Pichotta (if you say that fast, it sounds like SuperCoder, as Andrew points out).

Bob Gove also helped serve, after getting a carving lesson from Sabrina Guidroz. Andrew and Sabrina brought all the cooking equipment and supplied the culinary genius and most of the food.
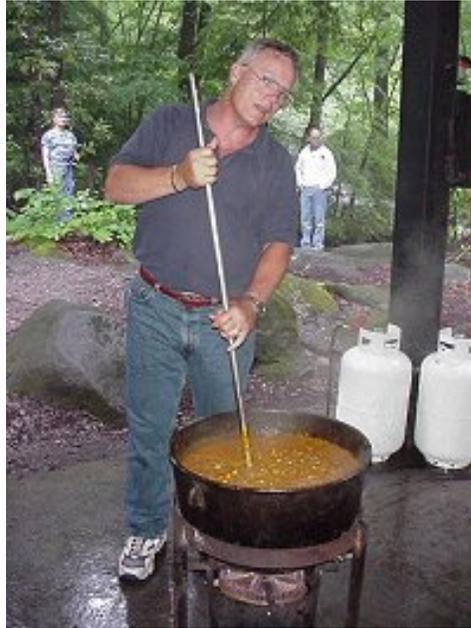
Larry Teames (left) took the day off to help with preparations, and Tom Hebenstreit took tickets (which guaranteed everyone got a full first serving).

Lee Hydechuk and Brad Noble (a.k.a. Team Beer Velocity) manned the beverage dispenser and did a whole lot more besides.

The Menu

I've been privileged to attend a few of Andrew's Cajun extravaganzas, and this one was the top. Aside from the perfectly done pig, the menu featured Cajun pure pork smoked sausage, boudin, crawfish etouffee over rice (tabasco available), French bread, corn maque choux, and for the vegetarians, green peppers stuffed with an eggplant dressing, baked sweet potatoes, and smothered okra with tomatoes, prepared by Master Chef Joseph (recipe by Sabrina). The legend grows.

More Photos And One Brain-Dead MPEG

We're hiring a different photographer next time. Really. Somebody who knows you don't shoot videos sideways. Here's Andrew giving the pig warning (MPEG, 416K). Those of you with rotating monitors and laptops not nailed to the desk, rejoice!

Two guys who really didn't want their picture taken (Bob Gove and Dave Harms).

Jim Mumford and Wayne Price: "That bear had claws, man! Like this!"

Andrew Guidroz to Jesus Moreno: "I don't care what the doctor said about your cholesterol, you get in that line and eat!"

George Hale explains to conference organizer Lee White what the boudin is *really* made from.

Brad Noble, Andrew Guidroz, and Lee White: "You're telling me that everyone here is either vegetarian or orthodox Jewish? What am I gonna do with all this crawfish and pork?"

Pete Halsted gets into the spirit of the evening: "Shake it up baby now!!! Twist and shout!!"

Sue Pichotta, her son John (a gifted musician, according to Lee's momma Dessie Wilson, who should know) and Arnor Baldvinnson. That could be Bob Dobbins on the right, but it's anybody's guess what he has under his arm. More boudin?
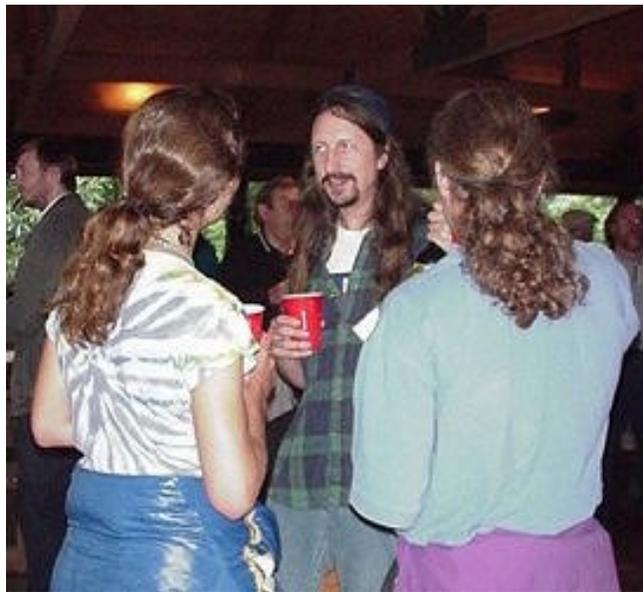
"Yeah, man, long hair is, like, cool. You dudes musicians too?"

[May 2000 News](#)
(May 31, 2000)

[Complete Cajun Cookout Coverage](#)
(May 30, 2000)

[Tool Talk: Vendor News from ETC](#)
(May 31, 2000)

[ETC Presentation Summaries, Part 1](#)
(May 31, 2000)

Reborn Free

CLARION online

published by
CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

Tool Talk: Vendor News from ETC

By Tom Hebenstreit, Reviews Editor

As I write this, I am cruising at 39,000 feet on my way back home after attending the second East Tennessee Clarion Conference (otherwise known as ETC).

Apart from the educational value of the conference (not to mention fun and frolic – and *boy* am I tired!), I also like to use these occasions to catch up with what some of the major Clarion tool vendors have been doing lately.

Mission accomplished.

Over the course of the week I had a chance to chat with most of the vendors who were displaying, so I figured I'd share a quick summary of those conversations and what I saw.

> **Note:** If something sounds interesting to you, please contact the vendor *directly* for more information. I cannot guarantee that anything you read here is flawlessly accurate – remember, I was talking and scribbling notes (actually, they look more like chicken tracks to me now) while simultaneously checking out screen demos. So these are first impressions, not in-depth analyses.
>
> In some cases, you'll already find information and possibly even demos of these products on the vendor's web site. Be sure to check them out by clicking on the link at the top of each section.

In any case, let me get out my patented ETC Super Secret Chicken Tracks Decoder Ring and begin this virtual tour of the vendor area.

Mike Hanson, **BoxSoft Development**

I have been a longtime fan (and user) of Mike's Super Template products, and always look forward to seeing what new wonders he has been cooking up. True to form, he has two new products being readied for release within the next four weeks or so.

Super Extras

First up is Super Extras, a brand new ABC-compatible toolkit built upon the base of the older Mike Hanson Templates set of free templates. New templates are being added, the old ones have been enhanced/expanded and the whole bundle will now be fully documented and supported. Over 30 templates are included. The cost will be US$99, and Mike expects to release the package before the beginning of July.

One extra note for current users of the existing Mike Hanson Templates -- if you use them with the Clarion 5.5 beta, there is a warning message that pops up during the IDE "Reading ABC Headers…" phase to tell you that a parent call will not work in one of the templates. I talked to Mike about this, and he said that while the parent call used to be valid in the ABC templates, it was never used. It sounds like C5.5 is just being a bit more thorough in checking the ABC template chain now.

In any case, he said he would be posting instructions on how to fix the templates on his website.

QuickBooks Export

This new product helps to integrate your Clarion programs with the popular QuickBooks package from Intuit (makers of Quicken, TurboTax and other products). As I understood it, this product will handle files created in the standard QuickBooks export formats. I'm not sure from my notes (darn chickens!) of the pricing, but Mike said it should be out by the end of June as well.

Miscellaneous Notes

Mike and I chatted about general C5.5 compatibility, and it seems most of his products have already been upgraded to work with the betas. He also mentioned that his Super PassCode templates (a set of registration/copy protection templates) had been expanded quite a bit in the last upgrade. He and Mitten Software (his distributor) are still considering putting together a bundle of Super PassCode and Super Limiter, Mike's templates for network user control and limiting. As the two products are quite complementary, it seems like a good idea to me.

Finally, he showed me a couple of new technologies that he had developed for a client that *may* become the basis for future products (cross your fingers, people). One was a template to easily create browse reports, i.e. drop in a template on a browse and you can have an instant good looking report from any browse that automatically picks up all current filter and order settings. From what I saw, this thing is *sweet*. He also showed me some pretty amazing work with what he labeled a flex-browse, where the browse changed dynamically based on both data and file relationships. If nothing else, these were certainly an impressive demonstration of how ABC and the templates can be extended.

Let's hope we do see some of this stuff as third-party tools later on this year.

Phil Will, [ProDomus](#)

Phil is the author of popular tools which enhance and extend many areas of Clarion, including file lookups (*PD Lookup* and *PD Drops*), date handling (*PD Dates*), finance tools (*PD Finance*), along with flagship internationalization tools such as *CwIntl*, *PD Worldwide Address Formats*, *PD Translator* and *Translator Plus*.

Receiving its first public showing was a new product called *PD 1 Touch Date Tools*. After Phil showed it to me, I'd have to say it really is well named. Phil dropped in one global extension template, and it automatically added default date types, popup calendars (including buttons for triggering them), date ranging, date field scrolling, localization and a lot more to the entire application. By the way, the popup calendars also support drag-and-drop to any date field (not just the one the calendar popped for).

Phil is also planning additional add-ons that will extend the date tools. One he showed me will add a daily calendar for scheduling, etc.

All in all, another impressive product.

I don't know the pricing yet, but the product is in late beta and Phil expects to release it officially within the next few weeks. For all you legacy users out there (myself included), yes, it will be available for both legacy and ABC.

Jesus Moreno, Gitano Software

With a wide range of add-ons and tools, Gitano not only has useful tools, but some of the best looking ones as well. From G-Reg Plus, their flagship registration and copy protection tool, through G-Calc, G-Cal, G-Notes and G-Buddy, Gitano consistently provides tools that not only work well, but look gorgeous.

G-Cal Pro

Although it has been a *long* time coming, Jesus was finally showing a late beta of G-Cal Pro. Designed to add calendaring and scheduling to your applications, this simple template and its accompanying library looked powerful, easy to use and had the usual Gitano visual flair. It has a wide range of calendar views, flexible multi-level scheduling and grouping options for appointments, and includes simple ToDo list functionality as well. Price is to be US$399 when it goes gold. Both ABC and legacy are supported.

A new feature recently added is the ability to generate HTML (i.e., web) pages for schedules and such directly from within G-Cal Pro.

As someone who has suffered through using ACT! and other personal information managers, I can only say that at first glimpse it looks like G-Cal Pro will provide scheduling that can go toe-to-toe with the big boys – and come out on top.

G-Cal

A utility to add simple calendaring and date functions to your application, *G-Cal* was recently expanded with two new calendars and a nifty date calculator.

Phil Carroll, Paragon Design and Development

Phil Carroll is the author of UltraJIMA (JPEG Image Management Automation) and UltraTree Platinum (the premier tree control template). We spent some time discussing the most recent additions to UltraTree as well as looking at planned features.

Phil has come up with a rather different way of pricing and releasing his products, in that new features are released as add-ons when they are completed, rather than waiting and lumping everything into a formal upgrade release.

You can either purchase the base product and then buy new features as desired (or

needed), or you can pay a bit more and get a premium support plan where all new features are included. In either case, bug fixes are always included.

Phil had added a number of new features since the last time I took a look at UltraTree Platinum, and all I can say is that this is one heavy duty product. Just about everything you can imagine doing with a tree control is there, ready and waiting to be used. Even so, there are currently nine more planned features listed on the Paragon web site. Yowza!

UltraTree Platinum pricing ranges from US$150 for owners of the prior UltraTree Profesional product through US$499 for a new UltraTree Platinum license with the Premium support option. It requires ABC, but for legacy users it also ships with the last legacy-compatible version of UltraTree.

George Willbanks, Banks Assoc.

George is the person who has taken on the unenviable task of corralling SQL guru Andy "Cowboy" Stapleton for a series of seven training computer based training (CBT) CDs on all aspects of using SQL and Clarion. After some delays, George said good progress is now being made.

Although the larger series is still under development, George has also had Andy record his ETC presentation (on moving from TPS to SQL and then the Web), and plans to offer that disc for US$99 (plus some shipping and handling, I would assume).

After watching some of the CD the day before Andy's presentation, I was tempted to ask when the English version would be available (Andy speaks Texan, in case you didn't know). All kidding aside, it looked like a great way for those who couldn't attend ETC to still gain the benefit of Andy's session.

There was also talk of getting some of the other presenters such as Dave Harms and Bruce Johnson to record their presentations as well. There is no current web site that I know of for these products, but you can contact George directly via email by clicking here.

Bruce Johnson, CapeSoft

Apart from showing the current versions of CapeSoft products such as the hugely popular File Manager 2, SecWin, MakeOver, EzHelp, and others, Bruce demonstrated what was to me one of the more intriguing new products at ETC.

Named NetTalk, this is a tool that implements TCP/IP communications within a set of templates and classes that look to be amazingly easy to use. In case you didn't know, TCP/IP is the basic protocol underlying the Internet, and native TCP/IP communications is normally not for the faint of heart.

NetTalk appears to have a wide range of capabilities built in. Indeed, Bruce says that their toughest problem right now is to stop adding cool features and just get the thing released.

Features already built in include file transfers, synchronizing date and time over networks, chat windows, the ability to create simple TCP/IP servers *and* clients, built-in DNS translation and a whole lot more. According to the CapeSoft Technical Briefing paper on NetTalk, they would like to add support for other basic Internet protocols such as SMTP (sending email), POP3 (receiving email) and FTP (file transfers using FTP servers).

Emphasis has been placed on robustness as well. According to Bruce, NetTalk can handle dynamic IP addressing, Dial-Up connections, and even cases where a connection gets dropped in the middle of a file transfer (*and* the IP address changes). He also states that the protocol supports packet checking for errors and error recovery.

Most impressive!

Currently in late beta, the announced price for *NetTalk* is US$299. You can buy it for US$199 if you purchase before it goes gold.

Summing It All Up

As Dilbert says: "I gotta get me some of that."

If I seem to have overused the word "impressive" in this column, I can only apologize by saying that most of the products I saw *were* exactly that – very impressive examples of the brilliance, ingenuity and above all, hard work of the vendors listed above.

Keep an eye out, as Clarion Magazine will definitely be covering many of these products in future in-depth reviews.

And trust me, you *should* have been there!

Reborn Free

CLARION online

published by CoveComm Inc.

Clarion MAGAZINE

TopSpeed

Clarion5 by TopSpeed

FREE Microsoft Internet Explorer

etc2000 EVENT SPONSOR

ETC Presentation Summaries

by Dave Harms

Keynote: Bob Zaunere

There's a separate article on Bob Zaunere's keynote address, as well as a screen shot of a press release which reiterates much of what Bob announced.

Dave Harms: Clarion, Linux, MySQL and Java - Managing A Multi-Platform Development Environment

I can't really give a critical summary of this presentation since it was my own, and Tom Hebenstreit, who was to review it, muttered something about keeping his job when I asked him to take notes. Ah well. The most important thing, at least from my perspective, is that it was done with early on, and I could relax. The second most important thing is that Bob Zaunere kindly took up a half hour of my time. I owe you one, Bob.

The two main points to the presentation were a) creating Clarion apps which access data on a MySQL Linux-based server, across a local network or the Internet (and with surprisingly good performance), and b) using Clarion, Java servlets, and MySQL to deliver web pages out of database (which is where ClarionMag is headed).

Setting up Linux, MySQL, Apache and Java servlets isn't a trivial task, especially if you take them on all at once. There's lots of help out there for each individual task, however. And once you get past all that, actually writing the servlets is quite easy, particularly if all they're doing is pumping out HTML from a database. It's really more work designing the database and creating the Clarion app to manage the database. So the trick is to find someone who's set up the Linux box for you.

Not all Clarion developers are writing web sites, of course, but a lot are looking at ways of creating web apps. MySQL adds yet another option by allowing you to connect a client application to a central database via the Internet. This is possible with other databases, but MySQL is designed with this in mind, and its access control system allows the administrator to restrict users by domain name and IP address (wild cards can be used). These restrictions can be made specific not just to databases, but to individual tables and columns as well.

MySQL is missing some of the features common to other databases, in particular support for transactions and triggers. What MySQL does have is speed and reliability. The presentation included a demonstration app which connected to a database in the CoveComm office in Winnipeg over a 28.8 dialup connection. Performance was good enough that several people asked later if that really was a 28.8 line. It was.

Much of the MySQL setup information can also be found in Clarion Magazine (search for MySQL), and I will be posting an update next month with some of the more recent information I presented at ETC.

Steve Parker: Thin Clients And Thinner

Steve Parker, one of the pioneer users of Clarion's web capabilities, gave a solid overview of Clarion web development, past and present, including Internet Connect, Java-Free IC, and of course Web Builder (and mentioning Michael Brooks' ClarionNet, Mike Pasley's Internet Framework Templates, and a variety of commercial thin client solutions). He also pounded the podium hard enough once to knock over the mike. We in the press notice things like that.

I was a little surprised at the time Steve devoted to the supposedly outdated IC product, but more surprised to learn that this is a real option for certain high-bandwidth deployments. IC apps have the great advantage of being dual-mode (they can be used locally and across the web) and take almost zero additional effort. IC really does still have a place.

Steve also briefly looked at Java-free IC, a template modification to improve IC's speed across the Internet. Mainly credited to Tony Goldstein, Java-free IC is a terrific enhancement to IC, but not the easiest to deploy. Although it might seem to be surpassed by Web Builder, in fact WB uses a lot of JavaScript and Java-free IC uses none, which is an issue if you're dealing with very old browsers.

Steve went over various app broker issues, including how apps share memory, the upside (lower memory usage) and downside (one corrupt app can bring down the broker and even IIS). Steve's server houses 30-40 apps from about a dozen developers, and despite the dire statements about app broker stability he did say that 98% of the problems he's encountered are in just two of his applications, and the problems are highly intermittent, coming in bunches. One attendee suggested that variations in the power supply could be a contributing factor.

One of the key points in Steve's presentation was his emphasis on how radically different coding for the web is from coding for Windows. The main difference, aside from screens delivered as pages, is that there is no ACCEPT loop running into which field level edits can be placed. It is possible to emulate ACCEPT with RefreshWhenChanged but it isn't a complete replacement. Coding styles have to be adjusted to compensate.

Steve also talked about various web app design issues, such as keeping menu sizes to a minimum, and using groups to force the creation of HTML tables, which you can use to control layout in some basic ways. He also covered Clarion's use of skeletons, and how the runtime system uses styles to determine which set of skeletons to use.

Steve has also written on these subjects for Clarion Magazine. See the author index under Steve Parker.

Nik Johnson: Using All Of The Clubs In The Bag

Nik Johnson's wide ranging presentation covered a lot of programming situations and problem solving techniques, not the least of which was a detailed explanation of how to use classes and templates to solve programming problems.

Clarion's great strength is and always has been generating code using templates. Nik covered the basics of the template language, and showed how easy it is to create simple templates that automate routine coding tasks. It's also possible (and effective) to use custom classes to add functionality to an application. A more advanced approach is to use templates to automate the integration and use of those classes. The class contains the logic, and the template gets programmer input and works with data defined in the application (fields, files etc). A template can also be used to create a class.

Nik gave some guidelines for class design and development, and showed a number of tricks, such as using classes as a substitute for local maps. He'd used local maps (which are declared inside procedures) successfully for a while, but when they began to fail he asked David Bayliss the reason and was told that local maps aren't fully implemented. But there is another way. A class declared in a procedure has procedure scope, and its methods can be used the same way a locally declared procedure would be used.

When creating classes, be aware that there are changes in declaration styles when moving from Clarion 5.0 to 5.5. As with class development in general, mimicking the ABC class library's approach is a good place to start.

Nik also reminded everyone of Barrington's Law: "that which is always required, should never be required." Don't duplicate code if you don't have to.

Skip Williams: Developing Outside Of The Web Edition Box

Skip Williams reprised his well-received 1998 topic, updated with recent developments. In a nutshell, Skip has created an Internet-capable application, not by using Clarion web development tools, but by taking his Binford 6200 chain saw and cutting the application down the middle, separating the user interface from the back end data source. He stitches the two halves together with FTP and HTTP, using wininet.dll (which is part of the Windows operating system) to exchange data between the client app and the back end app.

Skip's presentation was again an eye-opener, both in HTTP/FTP specifics, and in the general concepts of web development. He covered basics such as the difference between the GET and PUT HTTP requests, the various approaches to CGI (WINCGI and standard CGI), in which the browser calls a special purpose program which returns data. This data is usually text (and usually HTML, but it doesn't have to be for this kind of application), and what the CGI program sends back is typically dependant on the sort of parameters it receives. ISAPI works well for those using NT server and IIS.

The gem of Skip's session was the detail on using wininet.dll for HTTP and FTP exchanges. He demonstrated Clarion code to establish a session with a web server, request a page, and read that page. Similar code is used to initiate FTP file transfers. Skip also has a nifty web update process in which the user, with essentially one click, can check for any needed update, initiate the download of the first part of the update program (using Wise WebDeploy), execute the update, shut down the current application, and restart when the upgrade is complete.

Summaries Still To Come

- James Fortune: User Assistance 2000
- Andy Stapleton: Taking Sybase SQL To The Web
- Bruce Johnson: Understanding ABC So You Can Embed Your Own Code