

Reborn Free

CLARION
online

published by
CoveComm Inc.

Clarion MAGAZINE

Issue Index April 2000

[Main Page](#)

[COL Archive](#)

[Log In](#)
[Subscribe](#)
[Renewals](#)

[Frequently Asked Questions](#)

[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)

[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)

[Advertising](#)

[Contact Us](#)

TopSpeed

Clarion 5
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

[The Clarion Challenge: Shoot Yourself In The Foot Results](#)

The results are in! Read our Top 5 ways to shoot yourself in the foot with Clarion. (Apr 4,2000)

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

In the first of three articles on Clarion and MySQL, Dave Harms discusses installing Linux and the MySQL database server. (Apr 4,2000)

[Doodling Bitmaps](#)

Clarion does a reasonable job of handing graphics, but sometimes you need a bit more horsepower. (Apr 4,2000)

[ClarionMag Presents: Clarion Online!](#)

Clarion Magazine has obtained the rights to Clarion Online! Over the next several months we'll be republishing all of the articles published by COL during its two year existence. (Apr 11,2000)

[Using Clarion With MySQL Part 2](#)

In Part 2 of this three-partner Dave Harms explains how to configure MySQL permissions to allow a Clarion application to connect to MySQL running on Linux. (Apr 11,2000)

[Product Review: Special Agent 1.2 From CapeSoft](#)

CapeSoft's Special Agent templates make it easy to add interactive help to your apps using Microsoft's Agent technology. (Apr 11,2000)

[Cloning List Boxes Part 1](#)

There are many occasions when it is useful to have list boxes that are clones of each other. Jon Waterhouse updates a previous Clarion Online article with a new template and generalized classes. Part 1 of 2. (Apr 18,2000)

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

In the third and final installment of this series Dave Harms explains how to connect to and use/modify a MySQL database on a Linux machine. (Apr 18,2000)

[More Clarion Online: September '97](#)

The next Clarion Online re-issue is here! Read the September '97 issue. (Apr 18,2000)

[April 2000 News](#)

Clarion news, notes, and happenings from around the globe.
(Apr 25,2000)

[Cloning List Boxes Part 2](#)

There are many occasions when it is useful to have list boxes that are clones of each other. Jon Waterhouse updates a previous Clarion Online article with a new template and generalized classes. Part 2 of 2.
(Apr 25,2000)

[The Cranky Programmer](#)

Cranky points his patented CrankyScope at Clarion 5.5, Beta 2.
(Apr 25,2000)

[Next COL Update May 2, 2000](#)

The next Clarion Online update will be posted May 2, which is an off week for Clarion Magazine.
(Apr 25,2000)

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free

CLARION
online

published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)

[COL Archive](#)

[Log In](#)
[Subscribe](#)
[Renewals](#)

[Frequently Asked Questions](#)

[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)

[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)

[Advertising](#)

[Contact Us](#)

TopSpeed

Clarion 5
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

Shoot Yourself In The Foot Top 5

[Last month](#) we asked readers to contribute a Clarion version of the famous "How To Shoot Yourself In The Foot" piece. We received a number of excellent responses, and with some difficulty have reduced the field to the top 5.

[Drum roll](#), please.

Number 5

Clarion: After going to the Global extensions, setting the name of the GunClass, writing a BulletQueue management Routine, Populating the FootControl from a control template, and setting up the Actions tab of the ShootButton, you have a fully functional "Shoot yourself in the Foot" program. Of course, right before the bullet hits the foot, all things halt and a small window appears with the message, Error 47: Invalid Foot Declaration".

contributed by Josh Landin

Number 4

Clarion: You really want to shoot yourself in the foot and you used to know exactly where to do it, but now you can't find the proper embed point, so you pick several embed points to take a shot only to find you now have three holes in your foot arriving at different times. After checking the newsgroups you find someone wrote template that can shoot you in the foot anytime anywhere!

contributed by Parker MacDonald

Number 3

Clarion: You spend days creating a custom class to shoot yourself in the foot. When the class doesn't work, you spend several more days digging through the class libraries only to discover that you could have done it with one line of legacy code.

contributed by Rick Smith

Number 2

[The Clarion Challenge: Shoot Yourself In The Foot Results](#)
(Apr 4,2000)

[Using Clarion With MySQL - Part 1](#)
(Apr 4,2000)

[Doodling Bitmaps](#)
(Apr 4,2000)

[April 2000 News](#)
(Apr 4,2000)

Clarion: You shoot yourself in the foot and announce it on the newsgroups. Then everyone on the newsgroups complains that you should have been able to do it using COM or DCOM, the web-enabled version runs too slow, and there are bugs that prevent you from writing a program that shoots yourself in the hand, so they are going to use WinDev.

WinDev: It sounds like you could shoot yourself in the foot, but no one outside of France has ever actually tried.

contributed by Jay Guengerich

And The Number 1 Way To Shoot Yourself In The Foot With Clarion...

Clarion: It ships with gun, bullets, and a template that shoots. When you realize that you have to point the gun and pull the trigger all by yourself you get mad and flame Topspeed.

contributed by Jim Gambon

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

Multi-Platform Development: Clarion, MySQL, and Linux

Part 1 of 3

by Dave Harms

There's a lot of interest in Linux these days among Clarion programmers. Some are even asking for a Linux version of Clarion, and while that's not likely to happen any time soon, there may still be a place for Linux in your Clarion toolset as a database or file server.

In this article I'll look at the popular [MySQL](#) database and how you can use MySQL on Linux from your Clarion application. Although MySQL is available for Windows, I'll only look at the Linux version. There is a license fee for the Windows version, but not for commercial use of the Linux version unless, and I'm paraphrasing here, you make money from the sale or service of the MySQL server.

My exposure to MySQL on Linux came about because of Clarion Magazine, which is served by a Linux box running the Apache web server. ClarionMag uses of Apache's directory-based authentication to handle subscriber-only access; each directory is assigned an authentication group, and users (subscribers) are added to one or more groups. It's possible to use text files for the authorization lists, but once you get over a certain size of list these become quite slow as they aren't indexed. At first I used a flat file (DBM) database, but there are some problems with DBM files on RedHat Linux, and I wasn't happy with the results. It all came to a head when the user file reached 64k and the authorization database maintenance program (a Perl script) refused to add further records! I had one very intense afternoon installing MySQL and rebuilding Apache to use the MySQL database for authentication.

Moving to MySQL solved the database reliability problems, but the Perl script was still quite slow – it could take 20 seconds to update a single user's access. Since the rest of the magazine subscriber administration is handled by a Clarion application, it made sense to update the MySQL database the same way. All of the authentication database updating is now handled by a Clarion application, and I've been very happy with the results.

Clarion applications talk to MySQL databases on Linux using the [MyODBC](#) driver. The basic steps to getting this working are: install Linux; install MySQL on the Linux box; set up the MyODBC driver on your Windows machine; configure a data source, and then

[The Clarion Challenge:
Shoot Yourself In The
Foot Results](#)
(Apr 4,2000)[Using Clarion With
MySQL - Part 1](#)
(Apr 4,2000)[Doodling Bitmaps](#)
(Apr 4,2000)



Your source for
development tools
and add-ons.

Your outlet for
application sales.



May 23rd - 27th
REGISTER TODAY!

specify that data source in your dictionary; and configure MySQL on the Linux box to allow your Windows computer to connect. Provided you have a LAN/WAN or even an Internet connection to the Linux box, you're ready to use MyODBC tables in your application!

In Part 1 of this article I'll cover the Linux installation and setup, and MySQL installation. In Part 2 I'll look at using MySQL with Clarion.

Installing Linux

When I first set up a Linux box in late 1998 I didn't find it a particularly straightforward process. I muddled through the install without major problems, but in the finest *nix tradition, I had to recompile the kernel to add support for some of the features I needed.

Happily, Linux installations have come a long way since then. I recently set up a machine using RedHat Linux 6.1, and it was pretty easy. There are, however, a few concepts you should know.

Partitions, Directories And Mount Points

Linux filesystems are set up a bit differently from Windows filesystems. In Windows, you have a drive letter for each partition. In Linux you have directories instead of drive letters. But it's not quite that simple.

In the simplest Linux configuration you will need to create two partitions on your hard disk, typically during the installation process. One is a swap partition, which doesn't have a directory associated with it, and the other is the root partition, which is the top level directory. When creating this partition in the installation process you label it with a / (notice the forward slash rather than the Windows style backslash).

In most cases, however, you'll want at least one more partition, of no more than 16 MB, which will contain the operating system kernel and other boot files. This partition has the mount point /boot.

If you have a 500 MB drive, and a 100 MB swap partition, and a 16 MB /boot partition, that leaves 384 MB for the / or root partition. Both /boot and / are mount points, meaning when you switch to either of those directories (using cd / or cd /boot) you're switching to the corresponding partition.

But isn't /boot under the root directory? Logically, yes. But as soon as you drill down to /boot you're on a different partition because of the mount point. Any other directories under / which don't have a mount point specified will be on the root partition. Many Linux systems will have other partitions such as /usr, /usr/src, /tmp, /var, and so on. I use the following partitions on a 10 gig drive on the Clarion Magazine backup server:

| Mount point | Size |
|-------------|--------|
| / | 1 Gig |
| /boot | 16 Meg |
| /home | 2 Gig |
| /tmp | 1 Gig |
| /usr | 3 Gig |

| | |
|----------|---------|
| /usr/src | 1.2 Gig |
| /var | 500 Meg |

By using partitions and mount points you can restrict the amount of disk space available to any part of the directory tree. On a single user system that usually isn't a huge issue, however, so Linux installations for the desktop seem to be trending toward the single root partition (plus swap partition). Having a /boot partition is still a good idea when dealing with a very large root partition in the event that the BIOS is unable to handle the large partition size directly.

Window Managers

Another big difference between Windows and Linux is in the implementation of the GUI. In Windows the GUI is part of the OS; in Linux, and other *nix systems, the GUI is a separate. The Linux GUI is itself layered. First there is the X Window windowing system, currently in version 11, which handles GUI basics. Then you have a window manager, and over that a desktop environment. There are a number of window managers available, such as FVWM, AfterStep, and Enlightenment, and the two most popular desktop environments for RedHat Linux are Gnome and KDE. See <http://www.plig.org/xwinman/> for more information on window managers and desktops.

RedHat 6.1 gives you the option of installing a Gnome Workstation, KDE Workstation, Server, or Custom configuration. I went through the process a few times, just for kicks, and in the end I settled on the Gnome Workstation install. You might think the Server install would be the logical choice, but that's intended more for a machine that's to be a mail/web/news server. If you want all those things, go ahead and use the Server configuration. Just remember that the more services you have running, the more potential security holes you have to deal with.

By installing the Gnome Workstation I avoided installing a lot of unneeded services, and I also got a GUI. That was a help as I needed to download the MySQL software, and while I could have used the text-mode Lynx browser, I find Netscape just a bit more user-friendly.

Security Issues

Now that I've admitted to hooking my Linux box up to the Internet, I suppose a few words about security are in order.

It's very easy to use a MySQL database across the Internet. By default, the MySQL server listens for connections on port 3306, and if you have MyODBC installed on your Windows machine and you know the name or IP address of the MySQL server, you can in theory access your data from anywhere. And because of this MySQL has a fairly robust access/privilege system to control who gets in.

If you are going to expose your MySQL connection to the world, however, you need to think about more than just MySQL's own security system. You really need to install, or get behind, a firewall.

Even though the MySQL server has an access control system, you may want to limit attempts to connect to that port to only certain trusted machines. As well, almost any Linux box will by default have other services which are listening on other ports. These

can include telnet, ftp, http, news, mail, finger, name services, and more. You can strip out all the services which you don't want, but often a firewall is a better and simpler option.

The traditional way to create a firewall on Linux is with ipchains (or ipfwadm, in earlier versions). ipchains is a packet filter, which means it can inspect all packets which travel through any of your network adapters. You don't necessarily have to have a separate machine; you can run ipchains directly on the machine you're protecting.

Packet filters aren't the be all and end all of firewalling. All they do is inspect each ethernet packet and accept or reject it based on the rules you set up. These rules don't apply to the data in the packet, just to the packet header, in particular the packet type, the source, and the destination. You can use packet filters to allow/disallow packets from certain sources, or which are destined for particular ports.

The two common strategies are to allow all packets by default, and block packets on problem ports, or to deny all packets by default, and allow only the packets you know you want in. I recommend the latter approach. For more information on Linux firewalling see <http://linux-firewall-tools.com/>.

Getting To Know The Penguin

There are a lot of little differences between Windows and Linux that will probably drive you crazy for a while, if like me you have little if any *nix background. Using forward slashes instead of backslashes when specifying directories was easy to get used to, but going back to Windows was maddening. Whose brilliant idea was it to use that pinkie-wrenching backslash anyway? The ls command, equivalent (loosely) to dir, isn't all that descriptive when used without any options. And the text mode vi (or vim) editor is a bit of a throwback. Get yourself a good Linux book and be prepared for a learning curve.

Installing MySQL

Once Linux is running and you're moderately comfortable with its use, you're ready to install MySQL. The easiest way to do this, assuming you have your Internet connection established, is to log in as root, run the window manager (startx) and load up NetScape. Go to www.mysql.com and, to be nice to the MySQL folks, click on the Mirrors link and go to a mirrored site close to you. (But be warned – not all mirror sites carry all files!)

From the mirrored site go to the Downloads page. You'll need to follow two links from here; one for MySQL, and the other for MyODBC. First, look for the Unix Platforms link and click on the "recommended" release. As of this writing, it's 3.22.32. A portion of that download page is shown in Figure 1.

Figure 1. Download options for MySQL on Linux.

Note for RPM installations: If you are using the **RPM** to install, you should install at least the server and client **RPMs**.

Source downloads for 3.22

- [Tarball](#) (Including MySQL benchmarks and MIT pthreads).
- [Source RPM](#).

Note that this RPM does not work on RedHat 5.x systems because of glibc incompatibilities. Download the source rpm and do a rpm --rebuild to get a 5.x RPM

Standard binary RPMs for 3.22

- [The server for alpha systems \(RedHat\)](#).
- [The server for i386 systems \(RedHat\)](#).
- [The server for ppc systems \(RedHat\)](#).
- [Benchmarks/tests using Perl DBI for alpha systems \(RedHat\)](#).
- [Benchmarks/tests using Perl DBI for i386 systems \(RedHat\)](#).
- [Benchmarks/tests using Perl DBI for ppc systems \(RedHat\)](#).
- [Client programs for alpha systems \(RedHat\)](#).
- [Client programs for i386 systems \(RedHat\)](#).
- [Client programs for ppc systems \(RedHat\)](#).
- [Include files and libraries for development for alpha systems \(RedHat\)](#).
- [Include files and libraries for development for i386 systems \(RedHat\)](#).
- [Include files and libraries for development for ppc systems \(RedHat\)](#).
- [Client shared libraries for alpha systems \(RedHat\)](#).
- [Client shared libraries for i386 systems \(RedHat\)](#).
- [Client shared libraries for ppc systems \(RedHat\)](#).

Figure 1 shows links to various MySQL RPM install files (RPM stands for RedHat Package Manager). You'll need to download "[The server for i386 systems \(RedHat\)](#)" and "[Client programs for i386 systems \(RedHat\)](#)" and then run

```
rpm -i package-name
```

Install the server first, then the client. If you've been successful, you should be able to type

```
mysql
```

at the prompt and see something like Figure 2. Spend a bit of time with the MySQL documentation before you go any further.

Figure 2. The MySQL command-line client.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.22.32

Type 'help' for help.

mysql>
```

Last Resorts

If you have an older version of RedHat, or a non-RedHat Linux, you can look on the downloads page for a binary distribution or RPM. If that fails, you'll need to download the source code and compile MySQL yourself. It may sound a bit ridiculous to have to compile Linux programs before you use them, but in fact it's commonly done. Most Linux installations include a basic set of compile and make tools, and usually all you do is change to the directory where you installed the source, and type:

```
./configure  
make  
make install
```

Always check the README to be sure you're doing it right, however.

You have Linux and MySQL installed, and that's half the battle. In [Part 2](#) I'll explain how to set up permissions in MySQL.

[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.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked
Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To
Other Sites](#)[Downloads](#)
[Open Source
Project](#)
[Issues in
PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeedFREE
Microsoft
Internet
Exploreretc2000
EVENT SPONSOR

Doodling Bitmaps

by Jim Kane

How do you doodle?

I came to computers via a 20-plus year career in dentistry. In earning that degree I attended four years of high school, four years of college, and four years of dental school. With that background I feel very qualified to write about doodling! If nothing else, I learned at least that much sitting in all those lecture halls.

Then again, that was amateur doodling. I recently came across a situation that required something more professional. In this case, a company that owned a lot of scanning electron micrographs (SEMs) wanted me to develop an application that could not only display, zoom, and scroll the SEMs, but also overwrite one or more copyright notices. That meant drawing text and graphics on top of images.

I thought of using a string control and positioning it over the image, but the client rejected that idea as they wanted no chance of someone separating the image and the copyright notice. As well, the images were encrypted, but after decryption were device independent bitmaps or DIBs. Unfortunately, the Clarion image control does not support DIBs directly. After considering the options, I decided the Clarion image control was not a good fit for this project and set out to develop my own image control using the Windows API.

That may sound like an intimidating proposition, but you have to remember that the Windows API is the original tool for Windows programming. The Clarion language is a powerful layer on top of the Windows API but it can't do everything, and sometimes a little API is just what the doctor ordered.

About Bitmaps

Before I proceed a word about bitmaps is in order. Saying something is a bitmap leaves many options, as there are many types of bitmap. The broadest division is between device dependant bitmaps (DDB), device independent bitmaps (DIB) and the common BMP file format. A color DDB would look different on different computers. It does not contain a palette, just a color value stored for each pixel. It is completely up to the computer's hardware to interpret the color value. Because of this limitation, DDBs have

[The Clarion Challenge:
Shoot Yourself In The
Foot Results](#)
(Apr 4,2000)

[Using Clarion With
MySQL - Part 1](#)
(Apr 4,2000)

[Doodling Bitmaps](#)
(Apr 4,2000)

[April 2000 News](#)
(Apr 4,2000)



no disk file format, and are not exchanged between users. Their advantage is they are very easy to modify (or doodle on directly) with API commands. Because of this property, I will be dealing extensively with DDBs in this article.

DIBs contain a palette that keeps colors constant across various computers. DIBs have some associated API commands for manipulation, but not nearly as many as DDBs. When a DIB is stored to disk, a small file header (group structure) is added to the DIB and it is then copied to disk and stored as a BMP file. Normally, as a BMP file is read from disk, the file header is removed and a DIB is recovered, but Clarion instead chooses to leave the file header on and puts the DIB+file header into an image control.

For the remainder this article, I'll refer only to DDBs, and I'll call them bitmaps.

Handling The Bitmap

When Windows wants to refer to a particular window, it uses a handle or `hwnd` for that window. In Clarion you can easily get a handle for the current window with `hwnd = 0 {prop:handle}`. When Windows wants to refer to a bitmap, it also uses a handle, commonly called `hbitmap`. A bitmap is considered an object. As such there are functions available to create bitmaps, and a function called `DeleteObject()` that can delete a variety of objects, including `hbitmaps`. In addition to the ability to create bitmaps, the Windows API provides ways to get the `hbitmap` of existing bitmaps.

Before getting an existing bitmap or creating a new bitmap, you have to decide what device the bitmap will be displayed on. Typically this is either the video system or a printer. For this article I will deal only with video. Since bitmaps are device dependent, it is important to take into account the physical properties of the device the bitmap will be used with.

Device Contexts

The object that contains information about a device is called a device context, or DC for short. True to form, a handle for a DC is commonly called an `hDC`. In addition to holding the physical properties of a device such as the width and height of the display, dots per inch, and number of colors, a DC also holds all the tools used to display information. A bitmap when put into a device context becomes the canvas or paper on which you can draw. Drawing tools such as fonts, pens, and brushes can also be put into the DC and used to draw on the canvas, and the image can be scrolled/zoomed/repainted.

In summary, in order to display something on the video device the steps are:

1. Create or get a DC (`hdc`)
2. Create or get a bitmap (`hbitmap`)
3. Select the bitmap into the DC
4. Select other drawing tools as needed into the DC (Fonts, brushes, pens...)
5. Do some high class doodling using the tools on the bitmap.
6. Do any needed scrolling or zooming
7. Maintain the image on the screen by repainting as needed.
8. Destroy any objects created, and release any objects gotten

For the purpose of this demo, I thought rather than use a BMP or other image I'd do some screen capture and display a copy of the desktop in my new image control. The pseudo code is as follows:



```
!get the hwnd for the desktop
!step one done in a flash
Hwnd = getdesktopwindow()
Hdc   = getDC(hwnd)
!do other steps
!clean up and release what you got.
!Catch and release only.
releaseDC(hwnd, hdc)
```

Step 2 requires some thought. If I just grabbed the desktop image and started doodling on it, I'd be writing on the desktop itself! A better idea is to grab the desktop image and make a copy which I can display from within my own application. Before I can make a copy, however, I need to create a new bitmap to copy onto. Since copying requires the bitmap to be in a DC, I will also make a DC in memory. This is a common technique usually referred to as a memory DC.

So here goes...

```
!make another dc just like the one you have
!make a bitmap to fit in the dc
Hdcmem = createdcompatibleDC(hdc)
!make it the size of the desktop
!(code to obtain that is in the
! downloadable sample).
Hmbitmap = createcompatiblebitmap(hdc,width, height)
HOldBitmap=SelectObject(hdcmem, hmbitmap)
```

Since the memory DC has no physical device associated with it, I need to tell Windows what to make it like, hence the use of the functions with the word "compatible" in them. Lastly, for the bitmap to do any good, it has to be in a DC. `SelectObject` puts it there. Notice that whenever a bitmap is selected into a DC the previous DC (called `hOldbitmap` above) is returned so it can be saved and restored if necessary. It reminds me of the executive desk toy consisting of a row of steel balls on strings. If you pull one off to the side and release, after it collides another pops out the other side. DCs work the same way. They can only hold one of everything and when they are created have a default everything, so when a new something, in this case bitmap, is put in the existing one comes out. Keep the old bitmap that comes out because it will be used for cleanup like this:

```
Hmbitmap = selectobject(hdcmem, holdbitmap)
Deleteobject(hmbitmap)
```

Put the old bitmap back into the DC to get back the bitmap originally created. Once the bitmap is out of the DC, it can be deleted or destroyed. Failure to do so causes a minor memory leak. Of course, make that mistake in an inner loop and it becomes a major memory leak!

Blitting

Now for some fun. The thing that makes bitmaps so handy is `BitBlt` (which stands for Bit Block Transfer) and its related functions. `BitBlt` can copy a bitmap from one DC to another (this is called "blitting"). In this case I want to copy the bitmap in `hDC` starting

at X,Y coordinates of 0, 0 to a destination DC called hMemDC. In the destination DC I want the copy to start at 0,0 coordinates also. Since this is an exact copy (no scaling or zooming) the source and destination width and height are the same. Putting that together the code to copy a bitmap from hdc to hDCMem is:

```
BitBlt(hdcmem,0,0,width,height,hdc,0,0,ROPCode)
```

The ROPcode parameter gives various options for how the source color and destination color should be combined. Some possibilities include a straight copy, and various logical operations such as AND and OR. I will use just a straight copy which as every one knows is 00CC0020H. Well, good news, I made it through step 3 and I have my very own personal copy of the hBitmap (hMemBitmap) in my very own DC (hDCMem).

Being the nice guy that I am, I'd like to autograph the image of the desktop for you. The code to do this is in the [downloadable example](#), in the file called imgbmp.clw, in the procedure called Decorate. The code is very straightforward. I create a yellow brush with CreatebrushIndirect, then select it into the DC. Once the brush tool is in the DC, I use it to draw a rectangle. After that, I use the default font to print my message and name. Once done, I use the very same code as above to delete the brush and restore the DC to its pre-doodle condition. Please put some nice big checkmarks next to steps 4 and 5. Also since I have discussed the ever-popular topic of garbage collection (object disposal) I can take credit for number 8.

Before proceeding to the tricky stuff, take a look at the organization of the sample code. The file ImgBmp.Clw contains the example program. It contains the code for the demo window and for the region control upon which the improved image control is based. The code for steps 1, 4 and 5, which represents obtaining the initial image plus doing the customization, is there because it is likely to change from job to job. Most all the rest of the code (the tricky stuff), is tucked into a class called ImgDispCl for reuse. The class has one virtual method called GetPreferredSize(). The code in this method simply sizes the custom image control to be 5 pixels smaller than the window and centered on the window. Again that may change from job to job, so it is in the program module and not in the class. Darn handy those virtual methods – they let us customize a class for the job at hand. If you follow the example of ImgBmp.Clw you can adapt the class to your own needs without much difficulty.

Painting, Zooming And Scrolling

This only leaves painting, zooming, and scrolling. All these topics are controlled or happen in response to Windows messages. To get at Windows messages you need to tap into the message stream going between windows and our improved image control. The technique to do this is called subclassing. Fortunately I don't have to discuss it here as it was recently covered in a [Clarion Magazine article](#) by some dentist.

The code to enable scrolling all happens in response to WM_Hscroll and WM_Vscroll messages. When a Windows message is received, it comes complete with two long parameters called wParam and lParam. It's always somewhat amazed me that enough information to run Windows could be contained in these two parameters, but it is. For scrolling the wParam contains a constant that tells whether the user did a pageup, pagedown, lineup, linedown, scroll click or scroll drag. A case statement in the sample code processes each of these and calculates the new scroll position. Although the scroll position is also encoded in the Windows message it is only a 16 bit position, so in 32 bit windows it is better to call GetScrollInfo() to get the 32 bit position value. With that information the new X or Y position the user wants to move to is calculated.

Once you know where the user seems to want to go, you need to determine if this silly human really wants to go there. For example, if the image is already scrolled all the way down and the scroll bar gets a scroll down command, going further would display junk past the end of the image. In that case the move needs to be vetoed. To make this reality check calculation easy, when scaling and zooming the maximum amount the window can be scrolled is calculated and stored in `MaxScrollX` and `MaxScrollY`. The reality check then becomes just checking that these values are not exceeded.

Scrolling The Image

Once the scroll-to position is verified, there are two little details to take care of. First move the scroll bar to the new position. That is done by calling `SetScrollInfo`. This is very much like setting `prop:VscrollPos` in Clarion. Second, redraw the screen in the new scroll position. To do this call `InvalidateRectangle` to tell Windows what needs to be redrawn. The code is:

```
InvalidateRectangle(hwnd, 0,0)
-or-
InvalidateRectangle(hwnd,Address(Rect),0)
```

The second parameter tells Windows the coordinates of the area that needs to be redrawn. If the second parameter is 0, then Windows assumes the entire control needs to be redrawn. Since when the image is scrolled everything changes, I picked the first form and invalidated the entire window. Another option is to use `ScrollWindow()` or `ScrollWindowEX()` to scroll faster and minimize the amount of repainting, but I found the performance to be good enough without that. Or should I say I leave that as an exercise for the reader... Also note the mouse tracking code parallels the scroll code as the grabber hand does the same as the scrollbars.

The last parameter of `InvalidateRectangle` is probably the most important. If it is set to true, the invalid area is erased completely before it is redrawn. This is to be avoided as much as possible because it leads to a lot of flashing. In this case, since the new image will be the same size as the old image, erase is not needed. If the control was being resized this may not be the case and some erase is needed. Lastly, after `InvalidateRectangle` I call `UpdateWindow()` to tell Windows I want the repainting done *now*; otherwise it can take a long time to occur, and heck, I'm a busy guy. I cannot be waiting around. And besides it only takes one simple function call to get the faster repaint!

The `UpdateWindow` function sends out a `WM_Paint` message that tells Windows it's time to repaint. It would be wonderful if Windows would just do it but it does not. All Windows applications have to respond to `WM_Paint` messages and redraw themselves as needed. This is just one of the things that Clarion handles for the programmer, but when you're working with the API you have to do it all yourself.

All `WM_Paint` code starts with a `BeginPaint` and ends with a `EndPaint`. The paint structure can provide additional information to help optimize paints but I am not using them. Instead I use my old friend `BitBlt` to copy from the memory DC to the DC of the image control.

Zooming And Scrolling

On a zoom use `StretchBlt` instead to do the scaling. In connection with `StretchBlt`

the `Blt` mode needs to be set to `ColorOnColor`. This tells Windows to blend or dither colors to give the most natural scaling possible. If this step is forgotten the resulting image will surely be poor. It's something to keep in mind.

The last step is calculating the zoom and `MaxScroll` constants mentioned earlier. This is done in the `ResizeControl` class method. The first step is finding out the size of the window containing the control, and from that figuring out how big the image should be. This calculation is done without regard for the size of the image and may produce a control larger than the image. This will be corrected later. The virtual method `GetPreferredControlSize` does this.

Next the zoom factor is taken into account. Zoom calculations require multiplication and division of integers, which can lead to significant rounding errors. Windows provides a function called `MulDiv(A, B, C)` that takes care of rounding. It's a very attractive alternative to using reals or decimals since it is much faster. If the zoom factor is set to 50, the calculation returns the width and height of the image after zooming that will be half of the original image size. Maximum scroll values need to use that value, after adjustment for zoom, rather than the original image size. The values reflecting the after-zoom size are used by `StretchBlt` in the paint code to do the actual scaling.

Creating Scrollbars

If the image is larger than the displayable area you'll need scrollbars. If so, the scrollbars are displayed and the scrollbar range is set based on the zoomed size of the image. Although this is straightforward I must have made 100 errors before I got it correct.

The last section is the most critical. At this point I know what size the image control should be. My first impulse, as a Clarion programmer, was to use `SetPosition` to resize the image. When I did that, the screen flashed horribly. I could see my programming fee going down the drain fast. Motivated by greed, I did some experimenting and determined that `SetPosition()` forced a full erase of the image in conjunction with the move/resize of the image control. To avoid that, I instead used the `MoveWindow()` API call.

By setting the last parameter of `MoveWindow` to 0 so it did not erase, all flashing was suppressed and the control resized properly. The only problem was nothing repainted! Clearly I needed to invalidate something somewhere and call `UpdateWindow`. My first impulse was to invalidate the entire control and repaint the whole image. That worked but flashed a little since I was invalidating the entire image. When I resized the window by pulling down the lower border of the window the image enlarged and repainted properly, but when I shortened the window the control shrank, and the old image was still visible below the window. In other words my old, pre-resize image was now on the window outside the area of my control. Invalidating my entire control did not help, of course, since the old image was outside the control.

To deal with that problem I got the handle for the client area of the window (`{prop:ClientHandle}`). Using that `hWnd` and `InvalidateRectangle` I calculated the area of the window that needed to be cleared, and invalidated and erased that area of the window, and *only* that area. Since the erase was limited to the window below the control, it didn't flash at all. At last I had a nice smooth resize and repaint.

Well there it is: a custom control that meets the customer's requirements. Most importantly all the source code is available for modification. Since the image is readily available as an `hBitmap`, I can draw all I need to on the image. No matter what the

customer dreams up, since I can access the image while it is being displayed and have full access to the Windows messages, it can probably be done. And that is a good feeling.

[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.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

Clarion Magazine To Republish Clarion Online (COL) Articles

Winnipeg, MB - April 11, 2000

CoveComm Inc., publisher of Clarion Magazine, announced today that it has obtained the rights to all articles previously published in Clarion Online. Under the terms of the agreement, the COL articles will be republished in Clarion Magazine, and will be freely available to the public.

"We're very happy to be able to bring the COL articles into the public arena," said Dave Harms, Clarion Magazine's editor. "COL was in publication for two years, and that represents a lot of information, particularly for legacy developers."

The addition of the COL archive will more than double the Clarion Magazine knowledge base, and will be indexed by the Clarion Magazine search engine. The COL articles will be phased in over the next several months.

"There's some formatting to be done," said Harms, "but it's progressing nicely. If anyone is in urgent need of a particular article they can email me at editor@clarionmag.com and I'll do my best to move it up the priority list."

Harms also said that a PDF version of the COL issues would be made available after conversion was complete.

Asked if the original publisher of the Clarion Online articles profited from the transfer, Harms stated this was not the case. "They wanted the COL articles to go to a good home, and we're pleased they chose us. They provided the content, and we're providing the delivery."

Clarion Magazine is published four times per month by CoveComm Inc. Launched in February, 1999, ClarionMag has quickly become the leading web resource for Clarion developers worldwide. For subscription information see <http://www.clarionmag.com/policies/subscribe.html>.

[ClarionMag Presents: Clarion Online!](#)


(Apr 11,2000)

[Using Clarion With MySQL Part 2](#)

(Apr 11,2000)

[Product Review: Special Agent 1.2 From CapeSoft](#)

(Apr 11,2000)



Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked
Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To
Other Sites](#)[Downloads](#)
[Open Source
Project](#)
[Issues in
PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeedFREE Microsoft
Internet
Exploreretc2000
EVENT SPONSOR

Multi-Platform Development: Clarion, MySQL, and Linux

Part 2 of 3

by Dave Harms

In [last week's article](#) I covered installing Linux and MySQL.

In this article I'll go over some of the basics of MySQL administration, and next week I'll show how to connect your Clarion program to MySQL using MyODBC.

If you've installed MySQL from the binary RPMs as discussed in Part 1, then the MySQL utilities will be on the path, probably in `/usr/bin`. These utilities include `mysql`, which is the Linux MySQL client, and `mysqladmin`, which as you might expect is used for a variety of administrative purposes. These are both text mode applications, and although there are some graphical MySQL client and utility programs available for Linux (see <http://www.mysql.com/links.html>), I confess I haven't used any of them yet. In any case, you should know how to use `mysql` and `mysqladmin`.

Getting Started

Your first step to Clarion/MySQL success is to make sure the MySQL server is running. At the command prompt type

```
ps x | grep mysql
```

The `ps` (process status) command tells you what processes are active, and the pipe character runs the output through `grep` which only returns the lines containing the "mysql" string. You should see something like Figure 1.

Figure 1. Process status records showing `mysql` running.

```
404 ? S 0:00 (safe mysqld)
439 ? S N 0:05 /usr/local/mysql/ ←
                                libexec/mysqld
489 ? S N 0:08 /usr/local/mysql/ ←
                                libexec/mysqld
```

[ClarionMag Presents:
Clarion Online!](#)

(Apr 11,2000)

[Using Clarion With MySQL
Part 2](#)

(Apr 11,2000)

[Product Review: Special
Agent 1.2 From CapeSoft](#)

(Apr 11,2000)



If MySQL isn't running, type

```
safe_mysqld &
```

at the prompt to start the server. (Actually you shouldn't run this command as root, for security reasons.)

TIP: If you can't find a file you're looking for, try the `locate` command, as in `locate safe_mysqld`. If that doesn't work, run `updatedb` and try again. `locate` uses a database that typically is updated automatically once per day (via a `cron` job), so if you've just installed some software and you don't know where it went, you may need to run `updatedb` manually.

To see what databases are currently available, type:

```
mysqlshow
```

You should see a listing similar to Figure 2.

Figure 2. MySQL databases.

```
+-----+
| Databases |
+-----+
| mysql     |
+-----+
```

The `mysql` database is the administrative database. This can get a bit confusing, so pay attention to the case. When I write MySQL I'm referring to the product as a whole. Why I write about the `mysql` database, I'm referring to one particular database (collection of tables) which has the special purpose of controlling access to all databases available to MySQL on that server. Type:

```
mysqlshow mysql
```

and you'll see the tables in the `mysql` database, shown in Figure 3.

Figure 3. The mysql database tables.

```
+-----+
| Tables |
+-----+
| columns_priv |
| db          |
| func        |
| host        |
| tables_priv |
| user        |
+-----+
```

As you can guess from the table names, the `mysql` database contains information on which users are allowed to access which databases, tables, and columns. All user data will be contained in other databases which you will need to create. You should *not* add tables to the `mysql` database.

Controlling Access

The key to using MySQL is understanding its access control system. The three main tables involved are `user`, `db`, and `host`. All are contained in the `mysql` database. These tables represent a hierarchy of access control. In most cases you'll probably only need to work with the `user` and `db` tables.

Figure 4. The user table.

| Field Name | Sample Data | Purpose |
|-----------------|-------------|---|
| Host | localhost | computer the user is connecting from |
| User | root | User ID |
| Password | | Password (usually encrypted) |
| Select_priv | Y | Select records |
| Insert_priv | Y | Add records |
| Update_priv | Y | Update records |
| Delete_priv | Y | Delete records |
| Create_priv | Y | Create tables and databases |
| Drop_priv | Y | Drop tables and databases |
| Reload_priv | Y | Execute <code>mysqladmin reload</code> , <code>refresh</code> , <code>flush-privileges</code> , <code>flush-hosts</code> , <code>flush-logs</code> , <code>flush-tables</code> |
| Shutdown_priv | Y | Execute <code>mysqladmin shutdown</code> |
| Process_priv | Y | Execute <code>mysqladmin processlist</code> , <code>kill</code> |
| File_priv | Y | Read/write files on the server! |
| Grant_priv | Y | Give others same privileges as self |
| References_priv | Y | Not implemented? |
| Index_priv | Y | Create/drop indexes |
| Alter_priv | Y | Execute <code>alter_table</code> |

Any connection to the MySQL server involves at a minimum a host name (the computer from which you're connecting) and a user id. If you're connecting from the Linux box, then these will default to `localhost` and whatever user you're logged in as.

The root User

By default no password is set for the root user, as shown above. This means anyone who logs in as root on the MySQL machine has complete control over the MySQL database. Your first step in configuring MySQL permissions should really be to set the root password. Assuming you're logged in as root, use the command the command:

```
mysqladmin password mynewpassword
```

This will update the password field in the user table entry for `root/localhost`. You'd better write that password down in a safe place, because without it you'll have to reinstall the `mysql` database to get access!

Once you set the root password you'll need to specify it whenever you connect to MySQL. You do this with the `-p` option. For instance, if your password is "mypassword" then you'll need to call `mysqlshow` this way:

```
mysqlshow -p
```

and type the password when prompted, or type

```
mysqlshow -pmypassword
```

or

```
mysqlshow -password=mypassword
```

The last two options will display your password on screen, of course. Perhaps no one is looking over your physical shoulder. Even so, someone could be looking over your virtual shoulder. The `ps` command lists any currently executing processes along with their command line parameters, so in theory someone could telnet in, run `ps`, and see your password.

The rest of the columns in the user table control what kind of access this user has to the database.

The db And hosts Tables

Privileges granted in the `user` table are effectively super-user privileges and apply to all databases. That means if you grant Update rights to a user, that user can now update all tables in all databases, *including the mysql database*. Such a user is effectively a MySQL administrator and can change any and all information including root passwords etc. So be very careful about which `user` records have database privileges. You wouldn't normally have an application logging in as the root user, for instance.

In most cases users other than the root user will have all their privileges at the user level set to `N`. When MySQL receives an SQL statement and can't find a suitable permission at the user level, it next looks for permission in the `db` table.

The `db` table specifies privileges for a specific database. It's field structure is shown in Figure 5.

Figure 5. The db table.

| Field Name | Sample Data | Purpose |
|------------|---------------|--------------------------------------|
| Host | 192.168.100.1 | Computer the user is connecting from |

| | | |
|-----------------|--------|-------------------------------------|
| Db | testdb | Database |
| User | rocky | User |
| Select_priv | Y | Select records |
| Insert_priv | Y | Add records |
| Update_priv | Y | Update records |
| Delete_priv | Y | Delete records |
| Create_priv | Y | Create tables |
| Drop_priv | Y | Drop tables |
| Grant_priv | Y | Give others same privileges as self |
| References_priv | Y | Not implemented? |
| Index_priv | Y | Create/drop indexes |
| Alter_priv | Y | Execute alter_table |

As you can see db is similar to user except that it stores a database name instead of a user password, and it lacks the administrative privilege fields.

Messing With The Data

As installed, MySQL is accessible from the machine on which it is installed. To access it from another machine (i.e. one running a Clarion app) you need to set up some permissions. You can do this with the mysql client.

On the Linux box log in as root and type

```
mysql
```

or

```
mysql -p
```

if you've assigned a root password.

You'll want to create a database:

```
create database testdb;
```

You'll get a response something like:

```
Query OK, 1 row affected (0.01 sec)
```

When you start the MySQL client no database is selected. At the mysql> prompt type:

```
use mysql;
```

Actually the ; isn't necessary for this command but as you'll generally need the trailing ; for SQL statements it's a good habit to get into. MySQL will tell you the database has

changed. You can now update mysql and create the necessary records for your Windows machine to connect to the database you just created.

Let's say your Linux box and your Windows box are both on a local network, and the Windows box's IP address is 192.168.100.9. First, create a user record:

```
insert into user (host,user,password)
  values('192.168.100.9','rocky',
  password('squirrel'));
```

This insert statement specifies only three fields in the user file. All the remaining fields default to N, which again is a good idea unless you want to give this user administrative privileges. Note the use of the password function to do the encryption.

Add a db record:

```
insert into db (host,db,user,Select_priv,Insert_priv,
  Update_priv>Delete_priv, Create_priv,Drop_priv)
  values ('192.168.100.9','testdb','rocky',
  'Y','Y','Y','Y','Y','Y');
```

This record will allow the user rocky access to testdb when connecting from the specified host IP address (you can also use a host name). MySQL keeps the privilege settings in memory, so you have to tell it to reload this information from the database. Use the quit command to exit mysql, then run:

```
mysqladmin reload
```

For the complete mysqladmin command syntax see Figure 6.

Figure 6. mysqladmin usage: mysqladmin [OPTIONS] command command....

| Option | Description |
|----------------------|---|
| -#, --debug=... | Output debug log. Often this is 'd:t:o,filename' |
| -f, --force | Don't ask for confirmation on drop database; with multiple commands, continue even if an error occurs |
| -?, --help | Display this help and exit |
| -C, --compress | Use compression in server/client protocol |
| -h, --host=# | Connect to host |
| -p, --password[=...] | Password to use when connecting to server If password is not given it's asked from the tty |
| -P --port=... | Port number to use for connection |
| -i, --sleep=sec | Execute commands again and again with a sleep between |

| | |
|-----------------------------------|---|
| <code>-r, --relative</code> | Show difference between current and previous values when used with <code>-i</code> . Currently works only with <code>extended-status</code> |
| <code>-s, --silent</code> | Silently exit if one can't connect to server |
| <code>-S, --socket=...</code> | Socket file to use for connection |
| <code>-t, --timeout=...</code> | Timeout for connection to the <code>mysqld</code> server |
| <code>-u, --user=#</code> | User for login if not current user |
| <code>-V, --version</code> | Output version information and exit |
| <code>-w, --wait[=retries]</code> | Wait and retry if connection is down |

| Command | Description |
|------------------------------------|--|
| <code>create databasename</code> | Create a new database |
| <code>drop databasename</code> | Delete a database and all its tables |
| <code>extended-status</code> | Gives an extended status message from the server |
| <code>flush-hosts</code> | Flush all cached hosts |
| <code>flush-logs</code> | Flush all logs |
| <code>flush-status</code> | Clear status variables |
| <code>flush-tables</code> | Flush all tables |
| <code>flush-privileges</code> | Reload grant tables (same as <code>reload</code>) |
| <code>kill id,id,...</code> | Kill <code>mysql</code> threads |
| <code>password new-password</code> | Change old password to <code>new-password</code> |
| <code>ping</code> | Check if <code>mysqld</code> is alive |
| <code>processlist</code> | Show list of active threads in server |
| <code>reload</code> | Reload grant tables |
| <code>refresh</code> | Flush all tables and close and open logfiles |
| <code>shutdown</code> | Take server down |

| | |
|-----------|--|
| status | Gives a short status message from the server |
| variables | Prints variables available |
| version | Get version info from server |

Next: Connecting From Windows

MySQL should now be fully operational and ready to accept a connection from your Clarion app on a Windows machine. In the [third article](#) in this series I'll explain how to set up the MyODBC driver to work with Clarion.

[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.

Further Reading

There are a number of helpful MySQL articles at DevShed (www.devshed.com, an excellent site for anyone interested in web development):

The Beginning MySQL Tutorial

http://www.devshed.com/Server_Side/MySQL/Administration/page1.html

MySQL Administration

http://www.devshed.com/Server_Side/MySQL/Administration/

The MySQL Grant Tables

http://www.devshed.com/Server_Side/MySQL/Grant_Tables/

MySQL Documentation (also available on www.mysql.com)

http://www.devshed.com/Server_Side/MySQL/Manual/manfiles/

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free

CLARION
online

published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)

[COL Archive](#)

[Log In](#)
[Subscribe](#)
[Renewals](#)

[Frequently Asked Questions](#)

[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)

[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)

[Advertising](#)

[Contact Us](#)

TopSpeed

Clarion 5
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

Special Agent 1.2 From CapeSoft

Reviewed by Tom Hebenstreit, Reviews Editor

"Word has that cute little talking paper clip to help me," the customer says, "so why doesn't your program?"

What do you answer?

Sure, you can extol the virtues of pressing F1, using a search index to find a topic that never seems to be there, or wading through screen after screen of help that *almost* answers their question, but let's face it – in most cases the users want the help to come to them, not the other way around (darn lazy users!).

Or maybe they want a tutorial for your program: one that actually works within the program as it guides them through certain basic functions (just like Microsoft). Or maybe you just want a way to hold the users hand the first time they start the program without having to actually be there (just like Microsoft). Or you want to give your programs a self-running demo mode (just like...ah, you get the picture).

What to do, what to do...

You really have three choices:

1. Tell the user they can't have what they want (usually not a great idea).
2. Dig into the Microsoft Agent technology that powers features like the Microsoft Office Assistants, and write yourself an interface to it in Clarion.
3. Look into the CapeSoft Special Agent templates, and see if they can take the pain out of choice number 2.

Hmmm. Looks like option number 3 is the magic choice for today.

[ClarionMag Presents: Clarion Online!](#)
(Apr 11,2000)

[Using Clarion With MySQL Part 2](#)
(Apr 11,2000)

[Product Review: Special Agent 1.2 From CapeSoft](#)
(Apr 11,2000)

[April 2000 News](#)
(Apr 11,2000)

[COL: A Simple File Error Template](#)
(Apr 11,2000)

[COL: Complex Filters - When Parent Records Have No Children](#)
(Apr 11,2000)

[COL: Using Limited Related Drop Lists](#)
(Apr 11,2000)

[COL: Looks Like A Slider, Acts Like A Slider, Tastes Like Chicken](#)
(Apr 6,2000)

[COL: Oop In Clarion4](#)
(Apr 10,2000)

[COL: Quick, It's CWIC](#)
(Apr 10,2000)

[COL: Ask Dr. DePhobia - Questions and Answers about Clarion](#)
(Apr 10,2000)

[COL: The Joys of Reusable Code - Maintaining Your Own Function Library](#)
(Apr 10,2000)

[COL: Templates 101 - It's About Time](#)
(Apr 10,2000)

[COL: Under the Hood - Running Ahead](#)

Major Features

So what is Microsoft Agent? To quote Microsoft:

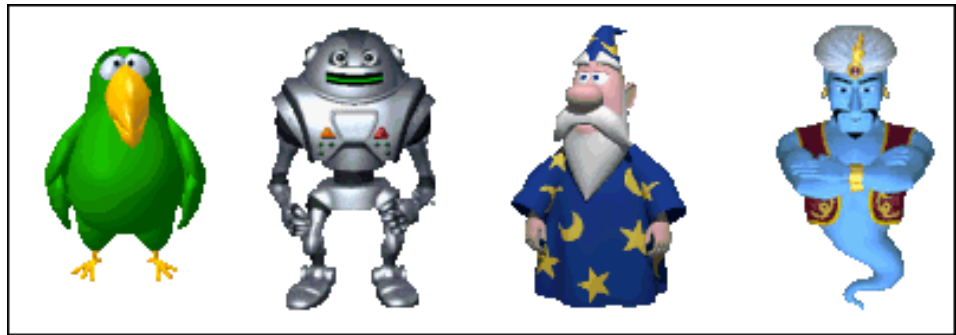
Microsoft® Agent is a set of software services that supports the presentation of software agents as interactive personalities within the Microsoft Windows® interface. Microsoft Agent's conversational interface approach is an extension and enhancement of the existing interactive modalities of the Windows interface.

Ummm...yeah, right.

Just think of Microsoft Office Assistants without quite as much juice (you don't expect Microsoft to give away their *best* stuff, do you?), wrapped up as an ActiveX component.

There are currently four available MS Agent characters, and Special Agent gives you access to them all (assuming you have installed them).

Figure 1: The current lineup of MS Agent characters



From left to right, they are known as: Peedy the parrot, Robby the robot, Merlin the magician and Genie (the genie, I guess). Microsoft also makes available an editor so that you can create your own animated characters (assuming you are that dedicated).

Some of the other highlights of Special Agent are:

- Templates are provided to simplify using most functions.
- A built-in point-and-click editor for creating, saving and editing the sequences of actions that drive your animated characters.
- The agent characters have multiple animations and sound effects already built in – you basically just string the various actions together.
- Supports text to speech, so your characters can not only display text balloons, but speak the words as well. Of course, they may sound more than a little like Stephen Hawking, but what do you want for free?
- And more...(see the links below)

Special Agent is compatible with Clarion 5, both ABC and Legacy. Clarion 5.5 support should be coming soon.

Installation

Special Agent actually consists of three or more installation procedures. At a minimum, you must install CapeSoft Special Agent (the templates, etc. for Clarion), Microsoft Agent (the ActiveX component which does all of the background work) and at least one

Agent (the animated character which interacts with the user). You can also download and install optional modules for speech generation and recognition, character editing, and more.

Going through each install, I found the following:

Special Agent (SA)

This install followed the Clarion 3rd Party Accessories (C3PA) guidelines, and thus it was both predictable and thorough. It found my version of Clarion 5, installed itself, added a link to the SA help to my C5 Accessories menu, registered the templates, and basically just took care of everything from A to Z without my having to do anything except click on the Next and Finish buttons.

As an aside, I install a *lot* of products and I've really come to appreciate the way C3PA compliant installs work. They do wonders for reducing clutter on the Windows Start/Programs menu and in making help easily available within the IDE via the Accessories menu. You already know what's going to happen before you even begin the install, and that is a good way to make customers happy. (If you are curious about the C3PA guidelines, you can find out more about them at <http://www.c3pa.com/standards.htm>.)

As for improvements, there were one or two little typos in the install. The install program also offered to update my AutoExec.bat file (which wasn't necessary) and, if I left that option checked, popped a message that the machine needed to be rebooted to finish the install (it didn't actually need to do that in my case). I'd like to see that option defaulted to No unless there really is a specific reason for it.

Microsoft Agent

After downloading the MSAgent.exe installation file from the CapeSoft web site, I ran it only to be immediately greeted with a message that "The Microsoft Agent 2.0 package is not compatible with the version of Windows you are running". Can you say "Huh?"

No matter, I figured, as I had just been given a MS Agent CD with Agent and all sorts of other goodies on it the day before at Internet World (this was directly from Microsoft). I popped in the CD, ran the program and – same error.

Hmmm. A quick run over to the Microsoft Agent web site (at <http://msdn.microsoft.com/msagent>) revealed that there was a new version of the MSAgent.exe install that worked with Windows 2000 (can you guess what I'm using?) It seems the problem is that most of the currently available Agent installable components (including MS Agent, the character editor, speech modules and foreign language support files) simply refuse to install under Windows 2000. They *work* just fine; it's just the installer that had the problem.

Microsoft has since released new installs which fix that pesky problem, so the moral of this story is that a) you should obtain latest Agent files from the MS web site listed above, and that b) CapeSoft should replace the versions on their site with the newer ones.

After all of this, how did the install go? Typical Microsoft. You must agree to a big old license agreement, then it puts the files where it wants to (in your Windows folders). Basically, one click and it was done.

Agent Character Files

Of the four available characters, I decided to try Robby the Robot first (ok, ok, so I have a soft spot for Sci-Fi and an aversion to things tainted by Disney – like parrots, Merlin and blue genies [and Mounties – editor].) Oops – I digress. Back to the install...

Installing the character was even simpler than installing MS Agent, i.e., no license agreement to agree to. You run it, it copies the files to a special location in your Windows folders, and you are done.

By the way, Robby doesn't look anything like the real Robbie the Robot (you just can't trust Microsoft, can you...)

Implementation

Now that the basics are installed and registered, how does it all work?

In a nutshell, your program talks to the MS Agent ActiveX control via a library of functions provided by CapeSoft. And how does it actually tell the character what to do? Via a "sequence," i.e. a pre-defined series of actions, such as look right, look left, blink, move over there and then say something.

This means that there are two things that must be done before Special Agent will be fully integrated into your application: first, you must add the templates and code to activate the character, and then you must define just what it is going to say and do in response to what the user does.

Let's tackle adding it to the app first.

Implementing Special Agent in your application

To begin with, you have to add a global extension template to the application. This is where you specify some basic information, such as the name of the file where your sequences will be stored, whether the Agent character will remain on screen after performing specific actions, and what hot key you will use to pop up the Sequence browser (more on this later).

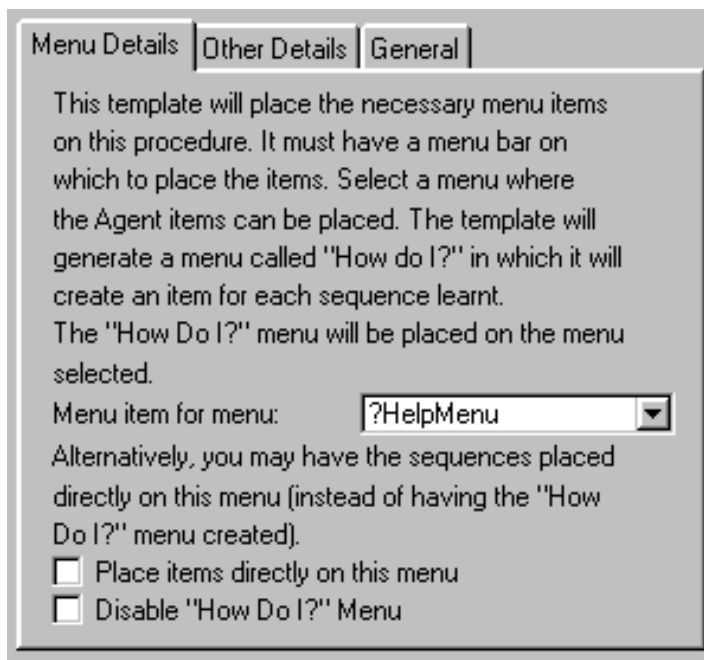
Once you have added the global extension and saved it, you'll find that it has also automatically added the Special Agent procedure extension to every procedure in your application. On your standard Windows and Forms, there is only one option on the local template, and that is to omit the Special Agent code for that procedure.

For certain other types of procedures such as Process and Source, the procedure extension code is automatically omitted.

The next step is to go to your frame procedure, where there are a few more special options to set for SA. For the most part, they deal with generating a "How do I..." option for the help menu (actually, any top level menu item you choose). This option will display a list of learned sequences (such as a tour, or specific guides or help), automatically generating them from your sequences file. Note that you can control which sequences appear on the list of items that the user can choose from.

While this all may sound a bit complicated, the actual template is very simple (as shown below)

Figure 2: The Menu Details tab for the application frame.



There are two options on the Other Details tab. The first is for specifying a sequence to run when the application starts, the second is a debug option that lets you see the events being generated on the status bar (assuming you have one on your application, of course).

And that's really all it takes to add Special Agent (and through it, MS Agent) to an application – setting a handful of options on two templates.

For added flexibility, there are some additional templates that make it easier to invoke the agents in specific places. They are:

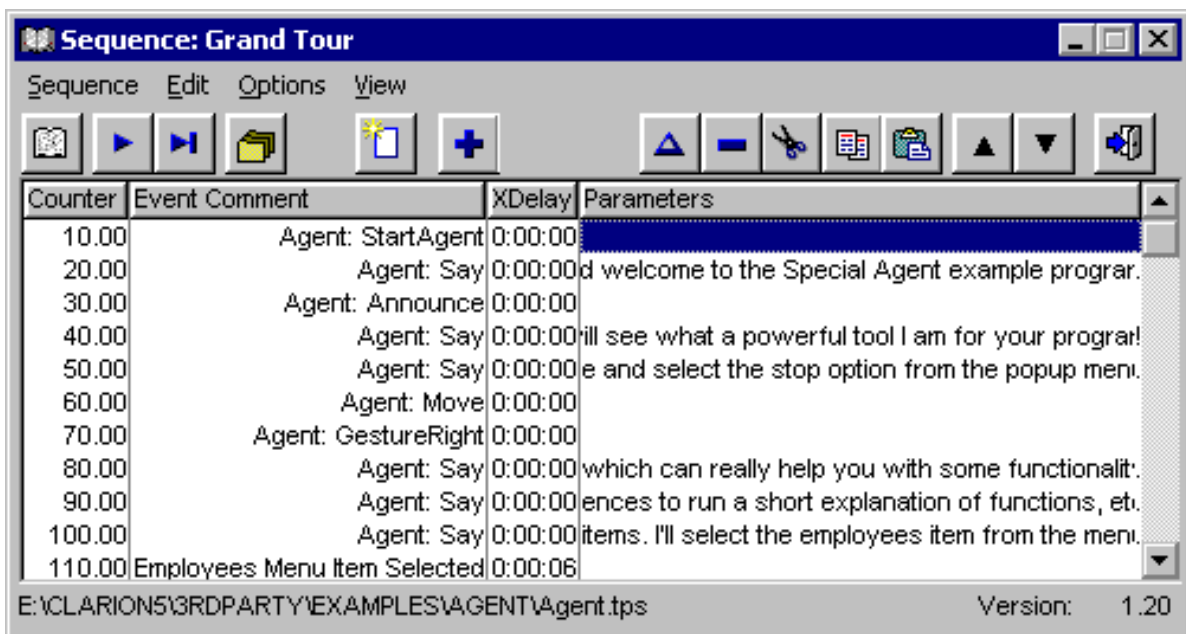
- A button template which pops up a browse containing the equivalent of the How do I... menu option. This button can be placed anywhere.
- A code template to call a specific sequence. You can use this to trigger a sequence from within any embed. For example, you might use this to begin a help sequence in response to a user error.
- A code template which can be used to simply pop up the Agent, have it say something and then go away (sort of a glorified Message() function).

For ultimate control, you can skip the templates and directly call the SA functions in your source code.

Creating Sequences

Once you have the basic templates in place and your application compiled, you begin the process of teaching sequences to your agent via a hot key. Control-F12 is the default (you can choose any key you like), and pressing it will pop up the Sequence browse (shown below).

Figure 3: The Sequence browse showing a Grand Tour sequence.



This is where the rubber meets the road, so to speak. It is from this window that you create, edit and link up all of your sequences. You create them by selecting various actions from a list, one step at a time, and then optionally typing in text if the action is to "Explain" something (shown on the list in Figure 3 as "Agent: Say"). You can also link sequences together, which is handy if you have a number of small specific sequences and want to create a larger "Tour" sequence or something similar. All sequence data is saved in a TopSpeed format file (which must be shipped with your application).

Special Agent does still have a few rough edges. For example, editing a sequence list can be a bit awkward, but I found I adapted quickly enough after a bit of fumbling around. Also, even though there are cut, copy and paste buttons (as seen in Figure 3), those features haven't been activated yet.

In any case, as I found out, implementing Agents is really like writing an interactive, animated help file. And just like a help file, creating something that really benefits the user can be much more time consuming than the basic technical requirements of putting it into your application.

Performance

What can I say? The characters and animations look great, on par with (or even better than) some of the standard MS Office Assistants. Everything was smooth and worked just as you would expect it to. I did notice that the MS Agent characters seemed to be a bit larger than their Office counterparts.

What they *do*, of course is another matter – and that part is totally up to you.

Documentation

Two documents are provided with Special Agent, both in HTML format. This makes them easy to read, easy to print, and the embedded links make it easy to jump to different sections of the file. The first file is approximately 21 pages, and is the basic Special Agent documentation that covers installation, implementation, tips, frequently asked questions, dealing with problems and so on.

The second file (which runs about six pages) is the technical reference for the base SA

functions that do the actual communicating with the MS Agent ActiveX component.

All in all, very good.

What about MS Agent itself? How about 24 MS Word DOC files covering *every* aspect of using agents (close to 700K zipped)? There are animation lists, docs for the character editor, information on how to use Agents on web pages, docs for the speech engines and a whole lot more. One of the most interesting ones that I found, though, was a non-technical sixteen-pager called "Guidelines for Designing Character Interaction" which did a good job of explaining the underlying concepts of why and when you want to use the agents.

Note that the Microsoft documentation (and indeed, all of the Microsoft Agent components) can be downloaded for free from the web site listed above and at the end of this review.

Licensing And Distributing MS Agent

MS Agent is free of all charges and royalties, but there are a few things to remember if you want to stay legal. If you plan on distributing MS Agent with an application, you must obtain a license to do so. Microsoft provides the license agreement along with the rest of the documentation (a mere nine pages of MS-speak), and you will need to sign it and send it to them in order to have permission to freely distribute their files (the MSAgent install, the character installs, etc.).

By the way, the MS Agent install can be run in a "quiet" mode by passing it a command line parameter. This allows you to run it from within your own install programs without the MS Agent agreement popping up in the middle of it.

Size-wise, Agent adds about 400K for the ActiveX install, and anywhere from 1.6 to 3.2 megabytes for each character depending on which one(s) you choose to distribute.

If you only plan to use it for yourself, you aren't required to obtain the extra license.

Technical Support

CapeSoft provides support via email, and has a sterling reputation for fast, comprehensive support. No surprises here.

What's Missing

This Special Agent doesn't quite have the spiffiest briefcase on the block yet, as it currently does not support all of the native capabilities of MS Agent.

Missing is foreign language support, though MS Agent itself has specific support for 29 foreign languages. Also missing (but less important, in my opinion) is support for drag and drop and VBX, OCX or OLE controls (i.e., it doesn't respond to events in those control types).

Summary






Love them or leave them, the MS Agent/Office Assistants are going to be around for a while. Why? Because many users like them. Sure, maybe *you* hate the annoying little paper clip creep and wish you could bend it into a pretzel every time it appears, but that doesn't alter the fact that you are building applications for *other* people, and part of your

responsibility is to take advantage of every tool you can find to make the life of your users a bit easier (not to mention reducing the number of support calls you have to handle).

Are agents appropriate for every task? Of course not. But if you want to use them in a Clarion program, CapeSoft Special Agent certainly makes it easy to get started. And there is no denying the extra pizzazz that having the interactive help provides.

Bottom Line: Since it is hard to really describe how these animated helpers actually look and act, the best way to evaluate Special Agent is to download the demo from CapeSoft's web site and give it a spin yourself. Try it; you just might like it!

Now where is my Special Agent Fox Mulder character?

| PRODUCT RATING | |
|----------------------------------|---|
| Overall |  |
| Ability to do the task | Very Good |
| Ease of use | Very Good |
| Ease of installation | Excellent |
| Documentation | Very 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 |  |

Source code for the Special Agent interface functions is not provided. As for MS Agent source code – what, are you kidding?

CapeSoft Special Agent lists for US\$199 and can be purchased via secure transaction through <http://www.clarionshop.com>.

For more information and a downloadable demo, visit <http://www.capesoft.com>.

For more information on the underlying Microsoft Agent technology, check out <http://msdn.microsoft.com/msagent>. All of the Agent components and documentation can be freely downloaded from this site.

Last Minute Update: CapeSoft informed me after the review was finished that they have updated their web site with the Win 2K compatible installs, released a C5.5 Beta 2 compatible version (free to all registered Special Agent users) and released a minor update to 1.21 which fixes some erratic editing problems in the Sequence browser. Busy guys, aren't they?

Vendor Comments from CapeSoft

Special Agent is currently on special for \$179 until April 30, 2000.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

Cloning List Boxes

By Jon Waterhouse

Part 1 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).

In an article in Clarion Online a while ago I presented a rudimentary class that allowed the user to add extra lists at will, move the lists, and drag and drop between lists.

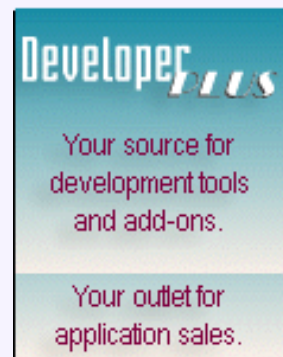
The major drawback of that class was that information about the box itself and the related queues and views had to be hard-coded. The usefulness of the code would obviously be greatly enhanced if one could specify the design of the boxes and their associated information in the Application Generator, with the class having only generic code.

In this article I'll discuss a combined class/template approach that meets this specification and allows the creation of multiple copies of similar lists, each one with a different range limit. The original approach also relied on a queue to manage new controls that was hand-written in the procedure. This function has been transferred to a class.

Think about how you would go about setting up a window with 12 of these lists using the existing FileDrop template (or browse template). You would have to add the 12 list boxes one by one, and on each one you would have to add the same fields in the list formatter plus any special attributes (width of fields, coloured cells, drag and drop IDs and any special queue formatting). The generated code would have 12 identical VIEWS declared and 12 identical QUEUES.

What the template described here will do is allow you to set up a single list box with all the desired bells and whistles and then, at runtime, use this prototype list box to create any number of clones of that list box. The attributes of the prototype will be applied to

[April 2000 News](#)
(Apr 18,2000)[Cloning List Boxes Part 1](#)
(Apr 18,2000)[Multi-Platform Development: Clarion, MySQL, and Linux Part 3](#)
(Apr 18,2000)[More Clarion Online: September '97](#)
(Apr 18,2000)



each new list. The `VIEW` will be declared only once, and the `QUEUE` only once (as a `TYPE`). Any embed code you add to the prototype list (e.g. in `SetQueueRecord` for formatting) will also be transferred to the clones.

Send In The Clones

My approach to this problem is quite simple: write a control template that is very similar to the template for a `FileDrop` list box, but with two important differences.

1. In addition to the `Init` method that initialises the properties of an existing class instance, there is also an `Addbox` method that clones a new instance of the class. It is in the `Addbox` method that the attributes of the box defined in the window formatter are associated with each new instance of the list box.
2. The regular drop list declares a `VIEW` and a `QUEUE`. The `VIEW` can be reused to populate the `QUEUE` in each new list box, but a separate (but identical) `QUEUE` for each list box is needed. The `QUEUE` should therefore be created as a `TYPE`, and each new list box will have to create (and destroy in the `Kill` method) a `QUEUE` of this `TYPE`.

First let's take a look at the two classes that are involved.

The `Multibox` class is inherited from the `FileDropClass`. Each `Multibox` consists of three controls: the list box; a region that can be used to move the list box around; and a title string, so that the list boxes can be easily identified. The five new class properties are the three field equates (FEQs) of the three controls, a value that acts as the range limit for the list, and the title string for the box. The class also contains modified `Init` and `Kill` methods, and a `TakeEvent` method that passes on the event to the appropriate derived method. `TakeListEvent` does nothing but provide a place for the programmer to specify what should happen on list events, and `TakeRegionEvent` handles dragging of the group of controls to a new location on the screen.

The second class is a `NewControl` class. The `NewControl` class keeps track of new controls (in this case `Multiboxes`) added to the window and passes any event on these new controls to the associated manager class (i.e. `MultiboxClass`). When you call the `AddBox` method one of the things `Multibox` does is pass the newly created list box along to `NewControl`.

In a standard program, a couple of `CASE` statements in `TakeFieldEvent` deal with events that need special handling. If you, the programmer, add code to be executed on the event on some field, the code generated looks like this:

```
CASE FIELD( )
  OF ?Myfieldequate
    CASE EVENT( )
      OF EVENT:SomeEvent
        Programmer code here
      END
    END
  END
```

When controls are added at runtime this approach is not available, since `?MyFieldEquate` is not known. What is needed is a method of passing control events on the new controls to the appropriate manager of the control. The way that the two classes combined (plus a few lines of template code) handle events is this:

1. At the top of the window manager's TakeEvent method the template inserts:

```
!NewControl handles all greater than regular controls
if field() > LastRegularField
  if not 0{PROP:AcceptAll}      !Ignore on acceptall
    r#=NewControl3.TakeEvent(field())
    RETURN r#
  end
end
```

so all events on new controls are handed to the NewControl class TakeEvent method to deal with. LastRegularField is set to LastField() when the window is opened.

2. In NewControl.TakeEvent() there is a queue (AllControls) that is sorted by field number. The record corresponding to FIELD() is retrieved from the AllControls QUEUE. This queue record contains the type of manager (Multibox) and the matching record in the Multibox queue. The same type of scheme is used by the WindowManager class, which maintains queues of FileDrops and Browses (although this scheme changes somewhat in Clarion 5.5 with the use of interfaces).
3. The Multibox queue record is fetched, which provides a reference to the instance of the Multibox class that manages the control the event happened on. The records in both queues were added by the NewControl.AddControl method when the Multibox was initialised.
4. The TakeEvent method of the particular instance of the MultiboxMultibox class is called by the NewControl.TakeEvent method
5. The Multibox class TakeEvent method uses the SELF.ListFEQ and SELF.RegionFEQ properties compared to the field number passed to determine which of the two methods, TakeListEvent and TakeRegionEvent should be called
6. TakeRegionEvent handles moves by moving all of the controls to the new location pointed to by a user's mouse move. The TakeListEvent is just a place holder to allow the programmer to enter code to deal with any events. Since the TakeEvent method of the parent class will be calling the derived TakeListEvent method in the child, this method must be declared as VIRTUAL in the parent class.

While the NewControl class given here has only the one extra queue (of MultiboxClasses), it could potentially have other queues that contain instances of other classes, whether standard ABC (e.g. browse boxes) or new classes.

The work that has to be carried out by the template is:

- Write the AddBox method of the Multibox class
- Deal with local data and class declarations
- Add code to the window manager's TakeEvent method that calls NewControl
- Get the Window Manager to call Init and Kill for the NewControl class.

Building the Multibox template is most easily accomplished by adapting the FileDrop template. The task of writing a template from scratch is daunting but,

despite the lack of documentation of the templates, other than on the language itself, it doesn't take too long to become familiar with the templates and figure out how to borrow from the existing templates.

Although the indentation and profusion of spiky # characters make it difficult to get the big picture of how the templates are arranged, each #CONTROL template follows roughly the same format. The `MultiBox` template (which is based on the `FileDrop` template) is no exception. The high level layout of the template is as follows:

1. The #CONTROL statement itself
2. A #PREPARE section that is run when you add the control for the first time in the application generator. This will create a default name for the new class object associated with the control
3. The CONTROLS section, which declares the controls that should be added to the window
4. The prompts section, which contains the code that provides the programmer with property sheets to modify the behaviour of the control. Some of the sheets may (like the Classes tab) be useful for more than one type of control. Thus some of this code may reside in a #GROUP, which is a section of reusable code.
5. Several #AT/#ENDAT sections. These specify statements to be generated in both the code and data sections of the module. Most of the sections have parameters that specify the particular ABC method that the code will be placed in. Most of the other #AT sections deal with code that will be placed in the data section. The #ATs included in the `MultiBox` template, in the order in which they occur, are:

#ATSTART is used to initialise template symbols

#AT(%GatherObjects) is used to add the template to the list of %ActiveTemplates so that templates can reference the objects belonging to other module templates

#AT(%DataSection), which is where the VIEW is generated

Next come all of the method #ATs. The method #ATs in the `MultiBox` template only reference method embed points declared in this template (%MultiBoxMethodCodeSection), but potentially #ATs could be added that places code in embeds declared in other templates. For example, the `NewControl` template puts code in a `WindowManager` embed.

The #AT(%LocalProcedures) section is immensely important. Here the embed points associated with the template are set up; if any of the embed points are used, the method source (as well as the preceding prototype) are generated. %ClassLines is updated to ensure that the local class declaration declares the locally derived methods.

#AT(%LocalClassData) is where the queue and other class data is declared

#AT(%LocalDataClasses) is where the class definition is generated. The generation relies on information put into the %ClassLines token to add any locally derived properties or methods to the class declaration.

6. Any #GROUPs that are needed. Not all the #GROUPs that are called or inserted in the main part of the template code need to be included in this template file — #GROUPs from any of the other template files can also be used.

In general, the #GROUPs that are specific to a particular template are included in the

template file along with the main template. Thus, if you want a section of code that builds a queue associated with a list box, the first place to look would be in the browse or FileDrop templates. In the FileDrop templates you find:

```
#AT(%LocalDataClassData)
#INSERT(%ConstructQueue)
#ENDAT
```

The %ConstructQueue GROUP itself is found in the ABBROWSE.TPW file. This GROUP will be used in a modified form in the Multibox template.

In addition to the template indentation making it a bit difficult to understand what is going on, the extensive use of tokens to hold scratch values, and many calls to #GROUPS in other files (particularly ABGROUP.TPW) also cause confusion. Keep the following in mind:


- %ValueConstruct is used as a temporary token. Many of the template commands will accept only tokens — they will not let you concatenate strings, for example. Therefore, first %ValueConstruct is #SET to the concatenated string, and then the token is used in another expression.
- %OOPConstruct is used in exactly the same way, but is generally used for class method and property declarations.
- The %MakeDeclr GROUP is used extensively to write data declarations. The arguments are:
 1. Start column of the second string
 2. The target string (the output line)
 3. The first string (which will start in column one, and will normally be the variable name)
 4. The second string (which will be placed starting in the position indicated in first parameter, and will normally be the variable type)

Often there are two calls in a row to %MakeDeclr. The second call adds a comment to the end of the line.

With that information in hand, take a look at the template in the downloadable source. Next week I'll explain the template details.

[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.



Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

FREE Microsoft Internet Explorer

etc2000
EVENT SPONSOR

Multi-Platform Development: Clarion, MySQL, and Linux

Part 3 of 3

by Dave Harms

If you've followed through the first two articles in this series you should have MySQL installed on your Linux box, and you'll be ready to get the Windows side working. And that means Clarion, ODBC, and MyODBC. Your Clarion application will use the TopSpeed ODBC driver to communicate with the ODBC layer, which will connect to the MyODBC driver, which will talk to the MySQL server on your Linux box, over TCP/IP.

From the MySQL downloads page (www.mysql.com/downloads, or one of its mirrors) click on Downloads for MyODBC. Choose the full setup install for the version of Windows you are running. Unzip the archive to a temp directory and run setup.exe. You'll see the window shown in Figure 1.

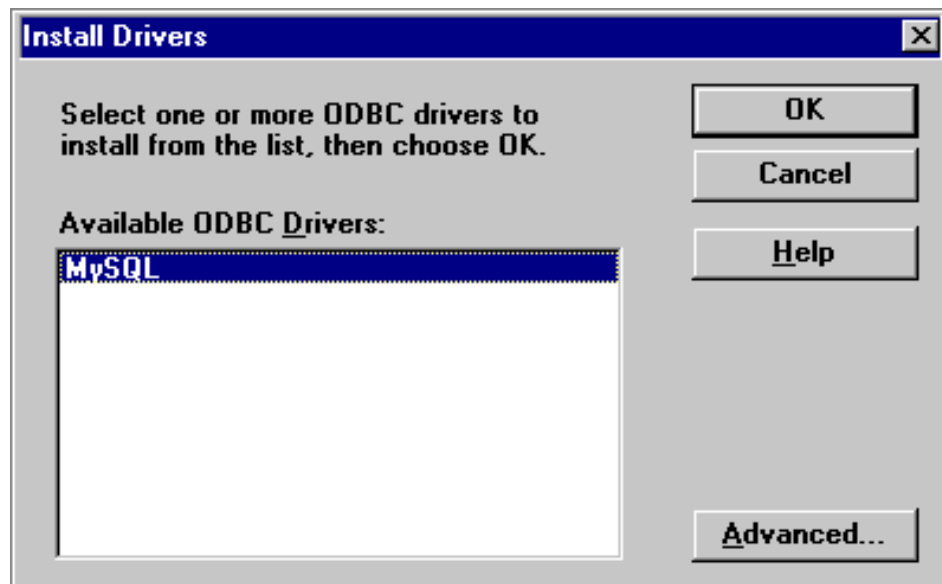
Figure 1. The ODBC setup program.



Click on continue and you'll be presented with a list of ODBC drivers to install, which in this case is just MySQL. Select the driver and continue.

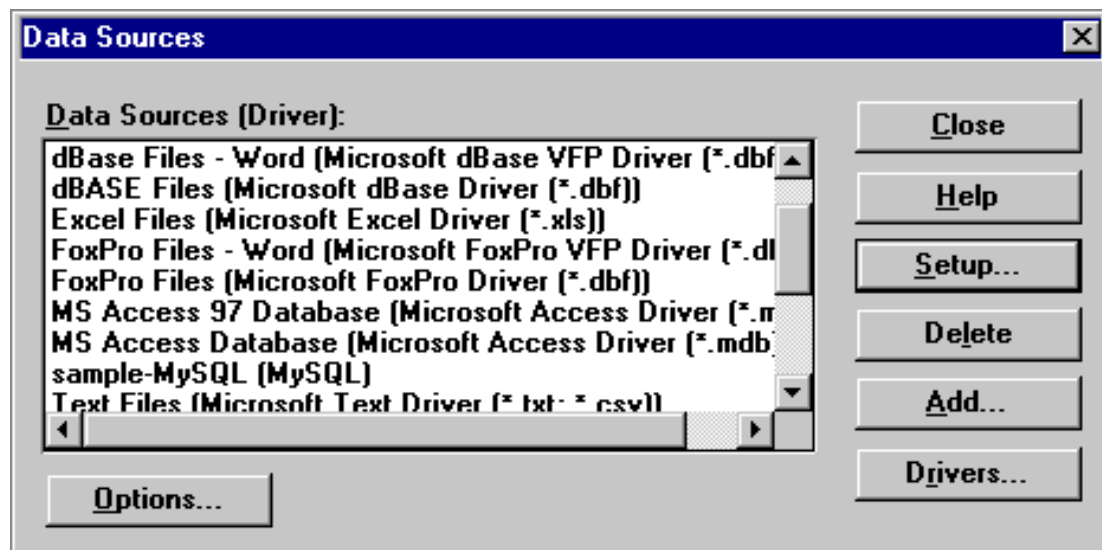
[April 2000 News](#)
(Apr 18,2000)[Cloning List Boxes Part 1](#)
(Apr 18,2000)[Multi-Platform Development: Clarion, MySQL, and Linux Part 3](#)
(Apr 18,2000)[More Clarion Online: September '97](#)
(Apr 18,2000)

Figure 2. Select the MySQL driver.



The ODBC setup program will add the MyODBC drivers to your system and register them with the ODBC administrator (see Figure 3). Finally, the setup program will display the administrator. You can set up a MySQL data source here, or you can do it later by choosing ODBC Data Sources from the Control Panel.

Figure 3. The ODBC Administrator.



You're Not Done Yet!

It could be that the version of MyODBC you download will work with Clarion, but it isn't guaranteed. I've had problems with several of the versions I've tried (typically an SQLBindParameter error), and the only version that as of this writing has worked for me is 2.50.25. Fortunately, once MyODBC is installed all you need to do is copy the appropriate MyODBC.DLL to your Windows system32 directory. If you want to try the 2.50.25 versions just download the zip using the link at the end of this article. You'll see two DLLs in the zip: MyODBC.DLL is has debug capabilities, whereas MyODBC2.DLL is optimized. If you want the optimized version just copy MyODBC2.DLL over top of MYODOBC.DLL (you'll first have to make sure you've shut down any applications which use MyODBC).

If you find another version of MyODBC that works for you, please email dharms@clarionmag.com and I'll see that a note is added to this article.

Data Sources

You'll need to set up at least one MySQL data source. In [part 2](#) of this series I set up a database called testdb. Figure 4 shows the configuration options for the MySQL ODBC driver.

Figure 4. Configuring a MySQL ODBC connection.

TDX mysql Driver default configuration

This is in public domain and comes with NO WARRANTY of any kind
Enter a database and options for connect

Windows DSN name: rocky-testdb

MySQL host (name or IP): 192.68.100.10

MySQL database name: testdb

User: rocky

Password: *****

Port (if not 3306):

SQL command on connect:

Options that affects the behaviour of MyODBC

| | |
|---|---|
| <input type="checkbox"/> Don't optimize column widths | <input type="checkbox"/> Pad CHAR to full length |
| <input type="checkbox"/> Return matching rows | <input type="checkbox"/> Return table names in SQLDescribeCol |
| <input type="checkbox"/> Trace MyODBC | <input type="checkbox"/> Use compressed protocol |
| <input type="checkbox"/> Allow BIG results | <input type="checkbox"/> Ignore space after function names |
| <input type="checkbox"/> Don't prompt on connect | <input type="checkbox"/> Force use of named pipes |
| <input type="checkbox"/> Simulate ODBC 1.0 | <input type="checkbox"/> Change BIGINT columns to INT |
| <input type="checkbox"/> Ignore # in #.table | <input type="checkbox"/> No catalog (exp) |
| <input type="checkbox"/> Use manager cursors (exp) | <input type="checkbox"/> Read options from C:\my.cnf |
| <input type="checkbox"/> Don't use setlocale | <input type="checkbox"/> Safety (Check this if you have problems) |

OK Cancel

Most of the fields are self-explanatory. The DSN name is the name you will need to enter in the Owner field of the Clarion file definition, and the MySQL host is the machine running the MySQL server. I've used an IP address here but you can also enter the machine name if your DNS is able to resolve it. To find out if the name is recognized you can try to ping the host, or you can try:

```
nslookup hostname
```

from the DOS prompt.

Enter the name of the database you're going to be working with, the

user you will log in as, and the password. And then you're probably done. MySQL communicates with outside clients via port 3306 unless the server has been told otherwise, so presumably you'll know if that's the case.

Importing Tables

If you have existing tables in your MySQL database, you can import them into your dictionary. Open the dictionary, choose File|Import File. Choose the ODBC driver, and when you get the Data Sources window, click on the 32 bit tab and choose the MySQL source you just created. You should see a list of tables on the MySQL server.

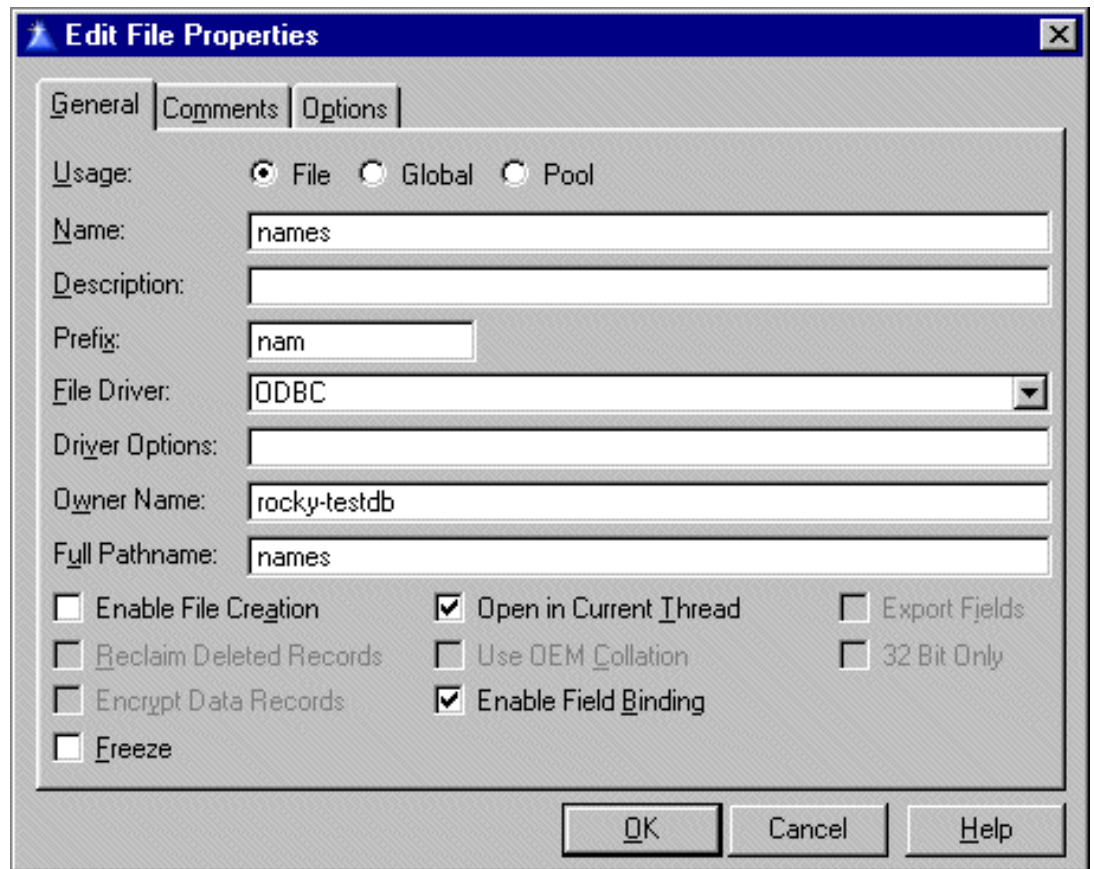
It's possible that you'll be unable to connect to the MySQL data source. Most likely this will be because you don't have sufficient permission to access the MySQL server. Remember that you need a valid user record as well as a valid db record in the `mysql` administration database. Also be sure you issue a `mysqladmin reload` command to make changes to the administration database take effect. See [Part 2](#) of this series.

If you still can't connect, verify that you're using the correct user ID and password. If all your data looks right, then it may be that you have a firewall between you and the server which is blocking port 3306, or there's some problem with TCP/IP.

Assuming you have succeeded in connecting, you can choose a table to import. Unfortunately you can't do a multi-file import as this requires the synchronizer, which doesn't work with the ODBC driver.

Figure 5 shows the properties for a table imported from a MySQL server. The owner is set to the ODBC DSN, and the pathname is the name of the table.

Figure 5. File properties for a MySQL table.



Creating Tables

If you don't have an existing MySQL database to work with, you'll need to create some tables in your own database. You can do this with SQL statements from the `mysql` client program on the Linux box (remember to issue a `use databasename` command from within `mysql` before issuing table creation/modification commands). If you have C5 Enterprise Edition you can use Data Modeler to create SQL scripts for you, but these will need some massaging before they can be used in MySQL. And I suspect there are some SQL utility templates out there that will do the job also, with minor changes.

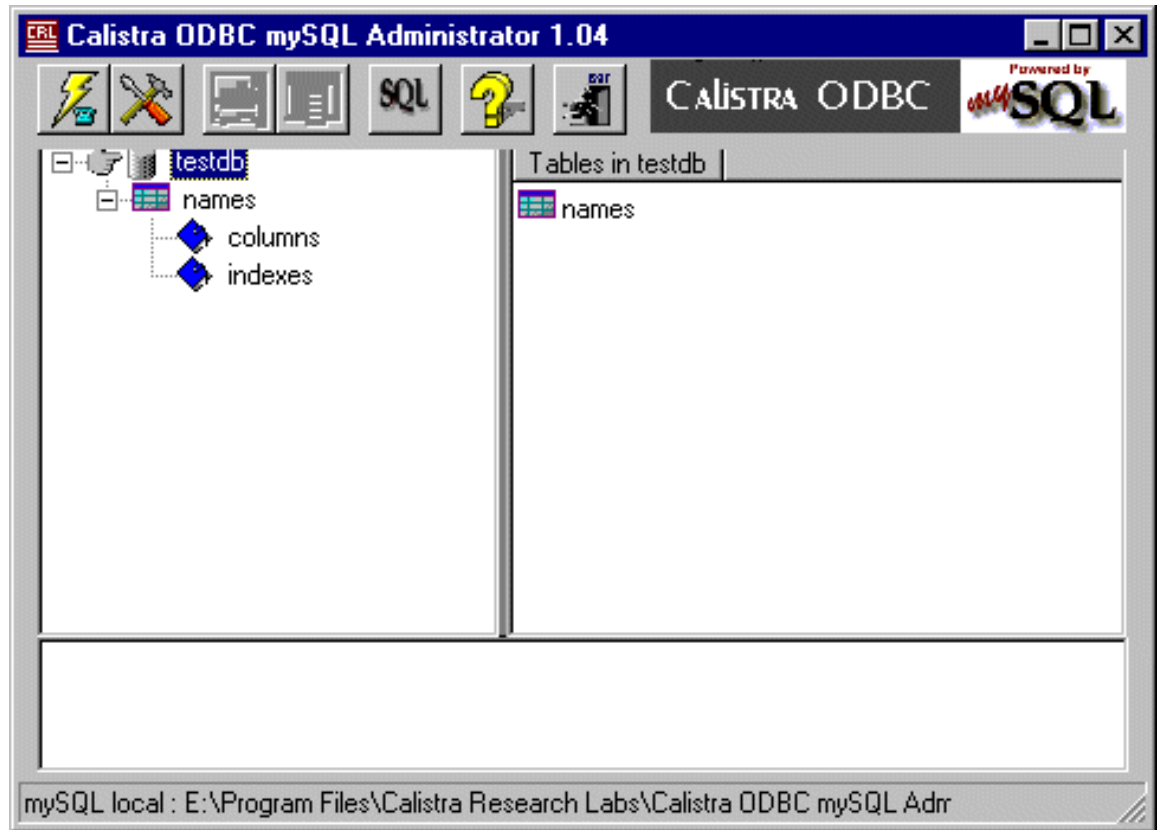
Another, often easier, approach is to create your tables using a free trial copy of the Calistra ODBC MySQL Administrator (COMA) from <http://calistra.com/MySQL/>. The free trial can be registered online for \$45.

COMA's user interface leaves a bit to be desired, but it's fairly functional. When you run the program, or when you choose the Connect button, you're presented with a list of all the MySQL data sources you've created. Once again, since data sources are really a combination of a database and user ID, and you want to use COMA to create/modify/delete tables, you'll need suitable rights as defined in the appropriate combination of user and db records in the `mysql` database.

Here's a tip, with a caveat. On the MySQL server create a user record with all rights and with your Windows machine's name or IP address in the host field. If you set all of the database creation/modification

privileges for the user, you don't need a corresponding database record. Then set up an ODBC data source and use COMA to make all other required changes to the mysql database. This way you only have to create the one user record by hand. And the caveat? You've just created a superuser record, which means that someone else could potentially log in (at least from that machine) and wreak havoc. So be careful what kinds of user records you create.

Figure 6. The Calistra ODBC MySQL Administrator



You can edit records in Calistra in two ways. For both you must be in data view. Choose the table you want to edit in the left-hand pane, then click on the View Data button (third from the left). To use edit-in-place just double-click on a field. To get a combination browse/form, right-click and choose Edit. On the left side of the edit window you choose the record to edit. The right side, shown in Figure 7, is the update form.

Figure 7. The update form portion of the edit window for the user table, from the mysql database.

Host: 207.161.247.82

User: rocky ... password

User Privileges

Select Update

Insert Delete

Admin Privileges

Create Shutdown

Drop Process

Reload File

Grant References

Alter Index

Notice that the user record shown in Figure 7 has no permissions set. Figure 8 shows the db record which gives this user access to the database. Remember, if no rights are found at the user level, MySQL looks for a database-specific rights setting. You can get even more specific by using the host table.

Figure 8. Rights assigned at the database level.

Host: 192.168.100.1

Db: testdb ...

User: rocky ...

User Privileges

Select Update

Insert Delete

Admin Privileges

Create

Drop

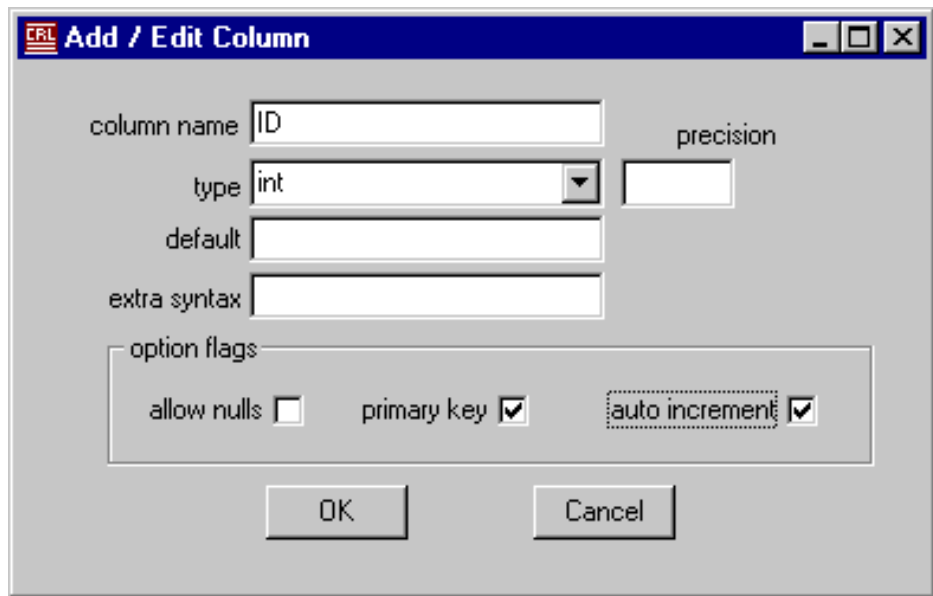
Grant References

Alter Index

Some Basics

Each of your MySQL tables should have a primary key on a field of type INT. This is easy to do with COMA. Just create a new field in the file, call it ID or something suitable, and check the Primary Key and Autoincrement options (see Figure 9).

Figure 9. Creating a primary key with COMA.



MySQL will take care of the autoincrementing for you if you choose that option, in which case you must not set the autoincrement option for this key in your Clarion dictionary. One important difference between MySQL and Clarion autoincrementing is that in Clarion the autoinc value is known while you're adding the record, but if it's done on the server it isn't. This will cause problems if you have an update form where you can add child records while adding a parent record.

Conveniently, if you decide to add an autoinc field after you have records in the database, COMA will automatically number the records for you. Very nice.

If you're changing MySQL and dictionary definitions in parallel and you get invalid record declaration errors when you try to run your app, import the table(s) into a different dictionary, verify that everything works with a quickie wizarded application, and compare definitions.

Summary

In my experience the biggest obstacles to using MySQL are getting the permissions right and finding a working version of the driver. At present, the 2.50.25 driver isn't listed on the MyODBC downloads page, but is available from ftp.mysql.com in the /pub/mysql/Downloads/MyODBC directory.

With those problems solved I've found MySQL to be as fast and reliable as its reputation suggests it should be. If you've been using MySQL with Clarion, drop me an email at dharms@clarionmag.com and tell me your story.

[Download MyODBC](#)

[Download the 2.50.25 DLLs](#)

Additional Information

Didier Le Duc recommends David Mansel's [MySQL Administrator for Windows](#). I've tried it, and I agree, it's an excellent administrative tool.

[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.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

Clarion News

April 25, 2000

[Four Weeks Left Till etc2000](#)

There are only four weeks remaining to register for etc2000, and rooms are going fast. Currently there are less than 25 rooms left in the reserved block. Once these rooms are filled there is no guarantee of accommodation anywhere even remotely close to the conference facilities as the conference ends on Memorial Day weekend and this is one of the busy weekends in the area. Conference fee is \$369.00, and for the small additional charge of \$50.00 you get to heckle Mr. Bruce Johnson in his full day class Saturday, May 27.

[LSZip and SetupBuilder Multi-Unit Discounts](#)

Linder Software has set up multi-unit discounts structures for the LSZip Data Compression Library (100% compatible with Pkzip and Winzip, provides a database in-memory compression feature) and Linder SetupBuilder 3.0 (16/32-bit installation system and binary patch builder). Discounts range from 20% for a quantity of 2 to 40% for a quantity of 50 or more. Evaluation Versions of LSZip Compression Library and Linder SetupBuilder are available.

[Free SQL Utility](#)

Innovative Management Systems has a free utility for users of MS SQL 7.0 or later. This exe (requires no installation) lets you view and/or edit database information including tables, indexes, stored procedures, views, and more. You can also execute SQL statements.

[Bug Reporting Tool Available](#)

Buggy 1.0 is a multiuser database program designed specifically for keeping track of bugs as well as feature requests (enhancements) and support requests. Supports multiple projects/software packages and multiple releases. A Security System is included to control who has access to Buggy. Exporting data to HTML and CSV document files, enhanced filtering and sorting on all fields and customizable charts/graphs are also included, as is a text search engine. A fully functional 30-day trial release is available.

[G-CalPro Beta Program](#)

Gitano Software plans to release G-Cal Pro by ETC 2000 time. A beta release is expected shortly, with subsequent beta releases weekly. Those who have already placed

[April 2000 News](#)
(Apr 25,2000)[Cloning List Boxes Part 2](#)
(Apr 25,2000)[The Cranky Programmer](#)
(Apr 25,2000)[Next COL Update May 2, 2000](#)
(Apr 25,2000)[Read The March 2000 News](#)



an order are invited to participate in the beta. In order to keep special pre-order pricing you do not need to take part in the beta, but you must fulfill your order by the time G-Cal Pro goes gold. For new customers the beta price will be \$299.00, and the final release \$399.00. If you are a registered user of G-Cal you are entitled to an additional \$49 upgrade discount. G-Cal Pro comes with the usual 30 day money back guarantee

[Boxsoft Templates For CW5.5 Now Available](#)

Updates have been released for Super Browse, Super Invoice, Super QBE, and Super Tagging. Super Security had been upgraded earlier (Nov99). Various bug fixes and some minor enhancements were also implemented. Other product updates will be available soon. These will be the last versions to work with Clarion4. Upgrades are available for a nominal fee.

[Clarion Development Direction Survey](#)

TopSpeed has posted a customer survey to gather input on future development. The rules for voting are simple: one serial number, one vote (valid serial numbers are any Clarion 4, Clarion 5, or Clarion 5.5). If you own each of these, you get three votes: one for each serial number. If you want to vote more than once, you have to run the app again and fill in all the required info each time. There are 40 features listed in the survey. Each Question has a hyperlink to open a popup browser window with more information about the question.

[Gitano Utilities Update](#)

New downloads are now available for GregPlus and GBuddy. A memory leak caused by C5.5 has been fixed.

April 18, 2000

[PD Date Tools Update](#)

Minor fixes to the One Touch Date Tools Beta 1 have been posted on the ProDomus web site. These are available for [C5](#) and [C5.5](#). These do not yet include the new Schedule Class. A new demo which makes limited use of the new Schedule Class is available for [download](#).

April 11, 2000

[Clarion Online Available At Clarion Magazine](#)

CoveComm Inc., publisher of Clarion Magazine, announced today that it has obtained the rights to all articles previously published in Clarion Online. Under the terms of the agreement, the COL articles will be republished in Clarion Magazine, and will be freely available to the public.

[Clarion 5.5 Beta2 Available Now!](#)

B2 has begun showing up in shops across the U.S.A. New features include the Fuzzy Matching template, the iBuild toolkit (including email/newsgroup messaging), CyberCash, and a shopping cart template. Call TopSpeed Sales at 1-800-354-5444 or 954-785-4555, or email sales@topspeed.com.

[Clarion Chat Page](#)

Craig Ransom has posted a collection of information about the various Clarion chats, as well as some brief instruction on the necessary software.



[Report-To-WMF Freeware Template](#)

Sterling Data has posted a free extension template which when used on a report will automatically create a numbered .WMF file of each page in the report. This template was kindly provided by Alex Oberhard of Komputery (Poland).

[Free Beta: JazzAge ActiveX Factory 2.5 Client Edition for Clarion](#)

JAZZAge Software has announced JazzAge ActiveX Factory 2.5 Client Edition for Clarion. JA CE 2.5 includes a visual Component Browser, coupled to a code generator, making it easy to browse the ActiveX components available on a site, display the complete description of the various classes/interfaces they contain, and then generate a source code module wrapping the complete selected component in the form of ready-to-use Clarion classes. JA CE fully supports the most advanced aspects of ActiveX client programming. Email info@jazzagesoft.com to receive a free copy of JA CE 2.5 for Clarion Open Beta 1.

[Ragazzi Supports C5.5 B2](#)

Software-By-Ragazzi's Developer's ToolKit, Templates and Utility Library (DLL) has been compiled and packaged for C5.5 Beta 2. Check the downloads page

April 4, 2000

[Wild Wild Ware Templates](#)

Wild Wild Ware has added a few more templates to an already extensive collection. Demos available.

[Clarion Essentials Class](#)

TopSpeed is currently organizing an Essentials class for Clarion 5.5 in Tampa, Florida that will be hosted by Russ Eggen. The cost of the class is \$1299. Attendees will also receive a free Interactive Self-Study Training for Clarion Foundations CD, a \$495 value. Seating for this class will be very limited. The class is tentatively scheduled for the second week in June.

[Clarion Essentials ISST CD](#)

TopSpeed's new Interactive Self-Study Training CD for Clarion Essentials should be ready for shipping during the first week of April. This CD picks up where the ISST for Foundations left off, and includes lab exercises and multi-media movies. Call TopSpeed Sales at 800-354-5444 or 954-785-4555 to reserve your copy.

[Technical Support Special For April](#)

TopSpeed is offering Clarion call-based technical support - normally \$120 per hour - for \$75 per hour. This special price is only available through April. Call TopSpeed Sales at 800-354-5444 or 954-785-4555.

[NetClip 3 Freeware](#)

NetClip3 is a handy little utility for copying Clipboard contents from one PC to another over a TCP/IP link. This means that it can be used over peer-to-peer, LAN, WAN, or even the Internet. NetClip3 has been developed in Clarion 5b using the ABC template chain and the NetTalk product from CapeSoft (www.capesoft.com), and compiled in Local mode so no external libraries are required. It is 32-bit and will run on Windows 9x/NT Operating Systems.

[Overlay Manager Update](#)

Mitten's Overlay Manager for Clarion Professional Developer v2.x(DOS) has now

updated for greater speed. A date problem in an OM4 procedure after Y2K resulted in slower compiles and no recognition of the latest procedure by Overlay Manager forced a total compile. This has been resolved and a technical update is available for purchase. This is the first update of Overlay Manager in almost six years. Contact sales.

[Easter Special At solid.software](#)

MessageEx and SysAni (both highly rated by Clarion Magazine) are available together for 59,- EURO (a savings of 19,- EURO). This offer is only available until Easter Sunday.

[ProDomus One Touch Date Tools Beta](#)

ProDomus has released a beat of its "One Touch" Date Tools, a major upgrade of PD's International Date Tools. Demo available.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free**CLARION**
*online*published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)[COL Archive](#)[Log In](#)
[Subscribe](#)
[Renewals](#)[Frequently Asked Questions](#)[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)[Advertising](#)[Contact Us](#)**TopSpeed****Clarion 5**
by TopSpeed

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

```

[April 2000 News](#)
(Apr 25,2000)[Cloning List Boxes Part 2](#)
(Apr 25,2000)[The Cranky Programmer](#)
(Apr 25,2000)[Next COL Update May 2, 2000](#)
(Apr 25,2000)



```

                                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 ABCCHAIN.TPL file. If desired, you *could* add to the code in ABCCHAIN.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 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 `%Mbrefer` token, and the setting of the `%FileDropObjectName` to `%Mbrefer`. 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(%Mbrefer,%ThisObjectName&'_ref')
#EQUATE(%FileDropObjectName,%Mbrefer)
...
#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 EXTRACTing 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 EXTRACTed 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 %OOPConstruct 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.

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.

Reborn Free

CLARION
online

published by
CoveComm Inc.

Clarion MAGAZINE

[Main Page](#)

[COL Archive](#)

[Log In](#)
[Subscribe](#)
[Renewals](#)

[Frequently Asked Questions](#)

[Site Index](#)
[Article Index](#)
[Author Index](#)
[Links To Other Sites](#)

[Downloads](#)
[Open Source Project](#)
[Issues in PDF Format](#)
[Free Software](#)

[Advertising](#)

[Contact Us](#)

The Cranky Programmer: Beta 2 – Woohoo or Boohoo?

After what seemed like an interminable delay, Beta 2 of Clarion 5.5 *finally* appeared in my mailbox. Was it worth the wait? Do the latest and greatest features justify all the marketing hot air expended on them? Is 5.5 Beta 2 going to rock my world, or just sock my wallet?

Given my always highly scientific approach to questions like these, I'll tell you what I'm going to do: over the next month or so, as time and space permit, I'll put Beta 2 under the patented CrankyScope and take a look at some of the new fixes and features.

To begin with I'll point the CrankyScope at the most visible new feature...

Visual Property Editors

For those who have not bought into the beta program, Clarion 5.5 has introduced a whole new way of getting to the various settings, templates and controls you use to build your applications. Named Visual Property Editors (I'll call them VPE for short), they offer fast track access into those settings in two major areas: the Application Generator main window, and the Window, Report and Dictionary formatters.

Figure 1: The App Gen window VPE (with most of the items collapsed).

[April 2000 News](#)
(Apr 25,2000)

[Cloning List Boxes Part 2](#)
(Apr 25,2000)

[The Cranky Programmer](#)
(Apr 25,2000)

[Next COL Update May 2, 2000](#)
(Apr 25,2000)

TopSpeed

Clarion 5
by TopSpeed

FREE Microsoft
Internet Explorer

etc2000
EVENT SPONSOR

Developer
PLUS

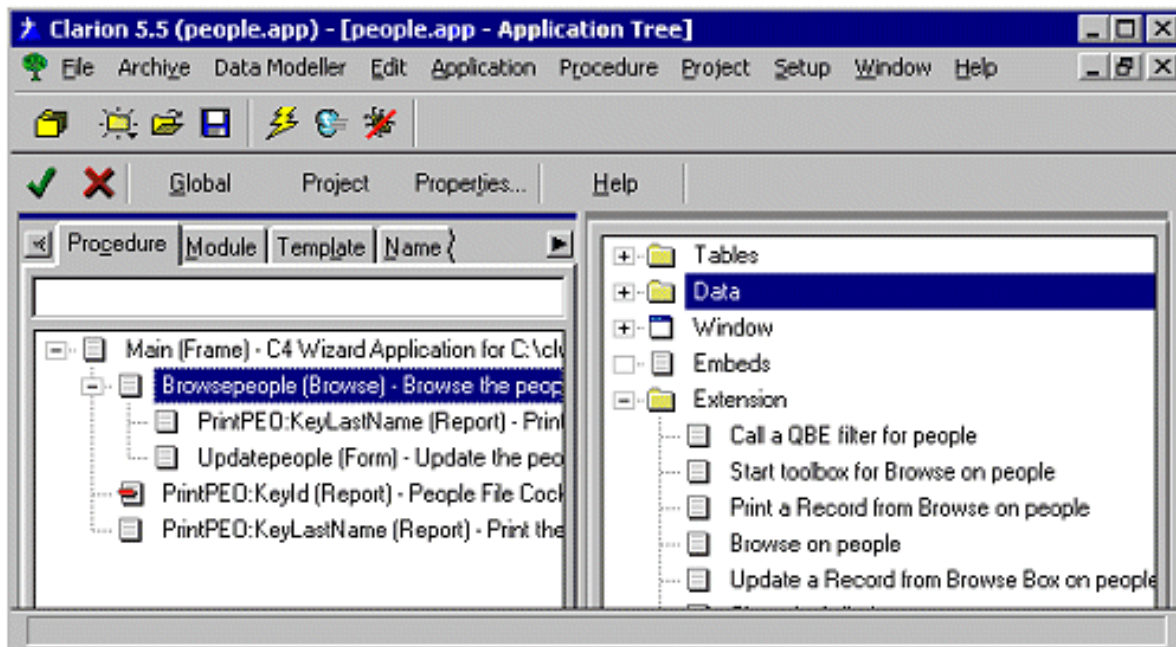
Your source for
development tools
and add-ons.

Your outlet for
application sales.

etc
2000

May 23rd - 27th

**REGISTER
TODAY!**



Clarion for Windows has always been a very vertical or drill-down oriented environment. By that I mean that everything always seems to be behind something else, and you constantly have to wade through long sequences of selecting things to get where you want to go.

If, for example, you want to change a button on a procedure, you have to select the procedure, then properties, then Window, then the button, then properties, then change a few tabs and finally, you get to the one property you wanted to change. To save your change, do the same in reverse.

The Property Editors are designed to be an antidote to this traditional Clarion click-a-rama, flattening out the IDE by exposing virtually every major procedure component right up at the top level. Not only does this greatly speed the process of making a simple change, but it also does wonders for giving you an overall picture of each procedure the moment you select it. Flatter, faster *and* more comprehensible – this is a good combination!

One of the biggest annoyances of Beta 1, however, was the way the Visual Property Editor in the Application Generator would constantly expand its tree every time you did something. For example, in Figure 1, you see that I have collapsed all of the categories except Extension (let's say I just want to check which extensions are currently in use on various procedures). In B1, every time you selected a new procedure or changed a setting, all of the other categories would fully expand again, pushing Extension back off the bottom of the window. Click – scroll, scroll, scroll. Click – scroll, scroll, scroll. Grrr.

In B2, the collapsed or expanded settings are preserved as you move around. Now *that's* more like it!

And now that they are working better, the Property Editors are one feature that I think will quickly become indispensable – especially since I believe TopSpeed has been taking some major steps *backwards* in usability over the last few releases.

Take, for example, when they switched to the single row scrollable tabs within the templates. With the previous stacked tab rows, it would take one click to get to the Totals tab for a browse template. With the single row tabs, you must click five times on the little right arrow button to reach the same tab. Further, the single row tabs don't

allow you to easily see what all of the available tabs are – you must scroll through them before you even know what options there are to choose from.

In my view, this unnecessarily complicated the use of the IDE (and that, of course, makes me cranky). See for yourself. Of the next two images, which one gives you more instant information (and easier access to the Totals tab)?

Figure 2: Stacked tabs – one click.

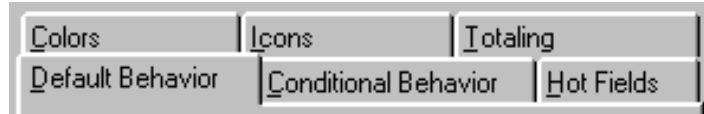
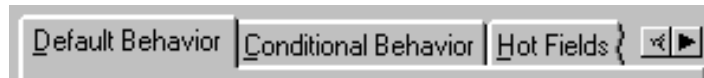


Figure 3: Single row tabs – click, click, click, click, click.



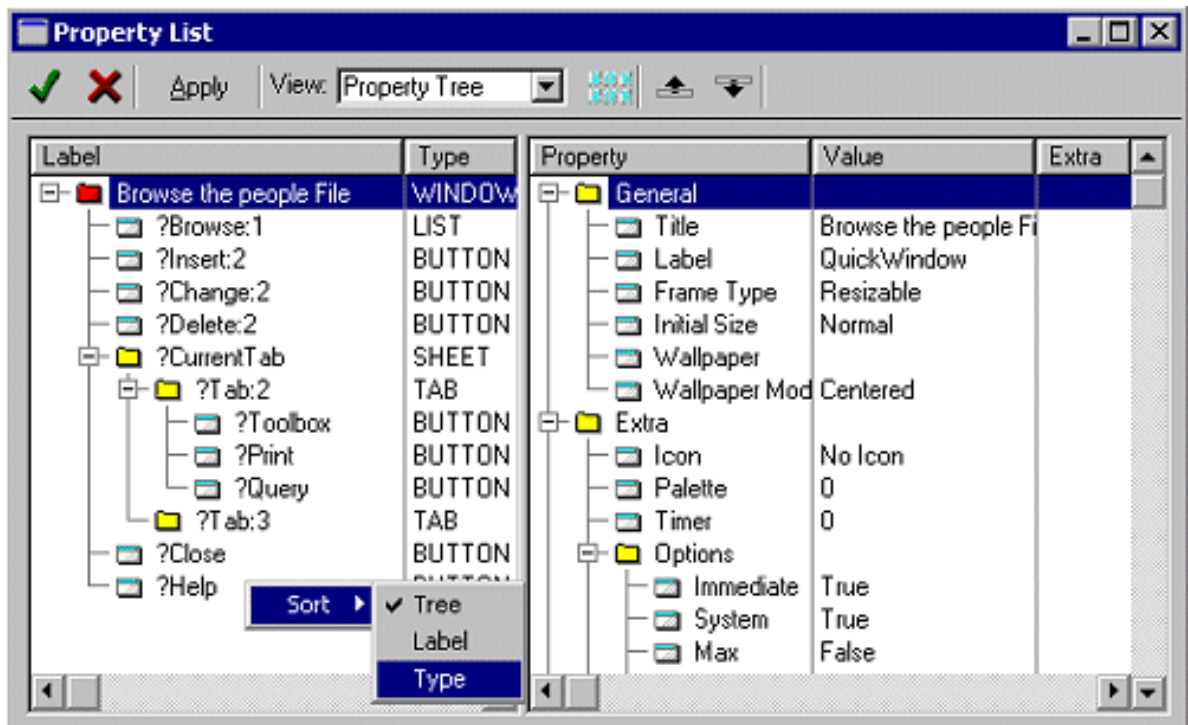
As I said earlier, flat is good. The more information you can expose at a high level, the easier it is to navigate and understand the program you are working with.

But I digress... Back to the VPE.

Where the new editors really start to shine is in the individual Window, Report and Dictionary formatters, and Beta 2 has made this good idea even better. To illustrate, take a closer look at the Property Editor within the Window formatter.

First off, where is the Property Editor? To bring it up, open the window formatter and either select Edit|Property Editor or press F12. You then see something like this (pay special attention to that little pop up in the lower left corner):

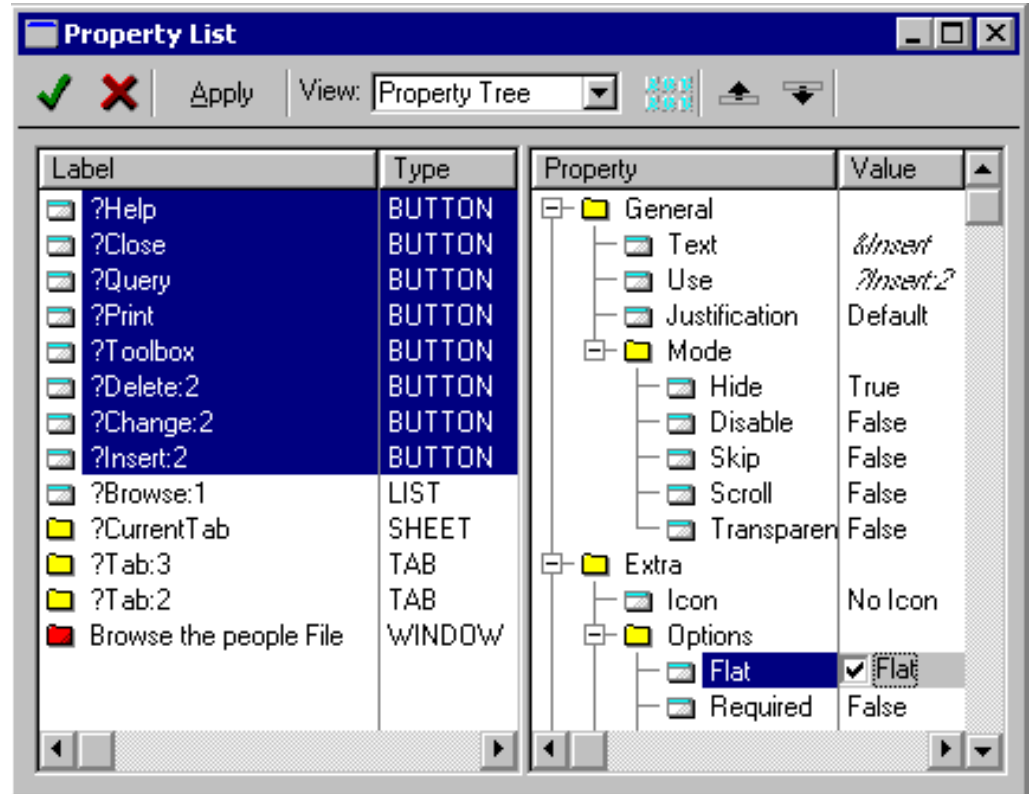
Figure 4: A right click brings up the sort options in the Property Editor.



Yes, you can sort the various columns (this doesn't seem to be mentioned anywhere in

the docs yet). You can also select multiple controls at one time, and apply the same change to all selected controls, as shown below in Figure 5. Another nice touch is that if you select multiple controls on the window before bringing up the VPE, those controls will still be selected there as well.

Figure 5: Sorted by Type, all buttons selected and changing to Flat.



To change all of the buttons on this window to flat, I have now gone from five clicks *per button* to five clicks for the lot of them. Hmmm, 40 clicks versus five clicks. Now that's what I call a measurable productivity increase (not to mention how much happier my poor mouse-abused fingers are!).

Note also in Figure 5 how some of the properties such as Text and Use are in italics. That tells you that those properties are different on some of the selected items (actually, all of the buttons in this case.) Properties that are the same for each button are shown as standard text.

I also like the Apply button you can see in the upper left corner of the VPE. You can make multiple changes, apply them and view the results, all without ever leaving the VPE. Sweet.

Could it be made better?

Is the Cranky Programmer cranky?

The biggest drawback so far is (you guessed it) the lack of an undo. Once you have pressed Apply (or closed the VPE and saved the changes), that's all she wrote. The changes are applied and the only way to undo them is to do it yourself. Oh well, at least it still takes less clicks to undo than it did before (provided you can remember the original settings).

Other things I'd like to see include the ability to sort simply by clicking on the VPE column headers, and for the property tree not to expand itself every time you select

another control. (Hey, that sounds familiar – how about fixing it here too, guys!)

Changing the order of controls is also better than it was, as the VPE now replaces the old Set Control Order option. Since you can select multiple controls and move them as a group, it is a large improvement over the old one-at-a-time method. It's still a bit quirky, though, as it seems to occasionally get confused and select additional controls after an up or down movement.

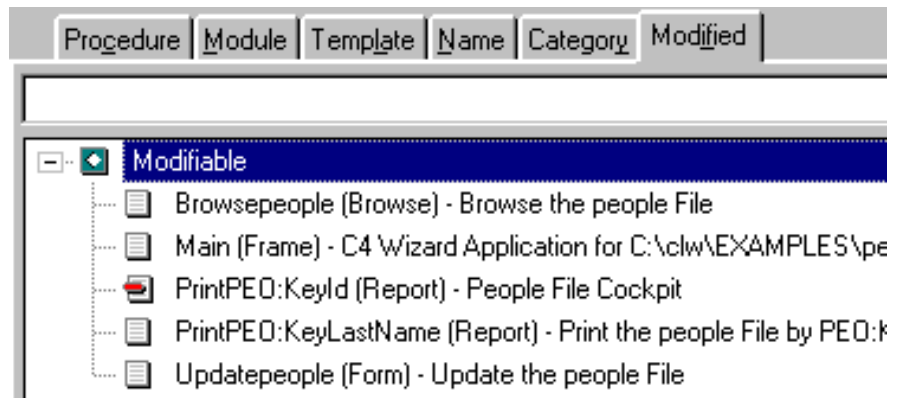
There is also a bit of a bug (ok, a big old bug) in there. If you select the alternate Property Column view (as opposed to the Property Tree view I have been discussing), choose a few columns to display and then exit the VPE, Clarion will GPF.

All in all, though, the new Visual Property Editors are one of those things that you'll soon wonder how you managed without. Of course, Microsoft has been using this type of tree-based property access for years, but we won't mention that.

The Modified Tab

The new Modified tab in the App Gen procedures window is still in the good-idea-but-not-quite-there-yet category. Why? Because even though it will sort the procedure list from most recently modified on down, it lacks one crucial bit of information: just *when* each procedure was last modified. Let's say I made some changes last week, and wanted to go back and check them out. Take a look at Figure 6. Can you tell which was modified last week versus which were changed the week before? Nope, you can't. And you *know* the information is available, or how could TopSpeed sort it that way?

Figure 6: The Beta 2 Modified tab.



Now, if it only looked more like this:

Figure 7: A crankified version of the tab.

| Procedure | Module | Template | Name | Category | Modified |
|------------|--------|----------|--------------------|--|----------|
| Modifiable | | | | | |
| | | | 3/27/2000 7:07 PM | Browsepeople (Browse) - Browse the people File | |
| | | | 3/27/2000 11:57 AM | Main (Frame) - C4 Wizard Application for C:\clw\EXAMPLI | |
| | | | 2/22/2000 1:55 PM | PrintPEO:Keyld (Report) - People File Cockpit | |
| | | | 2/22/2000 1:51 PM | PrintPEO:KeyLastName (Report) - Print the people File by | |
| | | | 1/25/2000 5:53 AM | Updatepeople (Form) - Update the people File | |

Better, isn't it?

Please, guys, just show us the date/timestamp. It won't hurt. Really, it won't.

Initial Impressions

So, is it worth upgrading to 5.5? Is Beta 2 enough to convince people to pry open their wallets? Well, on my own personal "I gotta get me some of that" scale, 5.5 is a definite improvement, especially since TopSpeed dropped that nonsense about not having a reasonably priced upgrade path for Professional Edition users. Thus, I'd have to rate it as a qualified "Woo" (hasn't quite earned the entire "Woohoo!" yet.)

I simply see a lot more basically *useful* stuff in 5.5 than I did in the 5.0 upgrade, i.e. things that I see an immediate application for in my own development efforts. (You know, I've still never actually *used* a Wizatron.)

What kind of useful stuff? Things like the new file driver enhancements (driver based triggers, better SQL support), new language statements (such as simple save/restore of file states, control cloning and interfaces) and new data types (the ever popular atomic Astrings, etc.), many of which I'll be touching on in future columns. Not to mention the real usability enhancements that I've talked about today.

So am I a happy camper?

Bwahahahahaha!!

Don't even get me started on the total lack of progress in OLE/COM support...

Copyright © 1999-2000 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited. If you find this page on a site other than www.clarionmag.com, email covecomm@mbnet.mb.ca.