



published by  
**CoveComm Inc.**

# Clarion MAGAZINE

## Search

## Home

## COL Archives

## Advertise

## Reader Comments

## Call for Articles

## Information

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

## Downloads

[PDFs](#)

[Freebies](#)

[Open Source](#)

## Site Index



Enter the [Clarion Magazine Sweepstakes](#) and you could win a **Compaq iPAQ** or an **ETC conference registration!** Other prizes too! No purchase required.

## PDF for September 23-29, 2001

All Clarion Magazine articles for September 23-29, 2001 in PDF format.

*Posted Monday, October 01, 2001*

## Review: LogFlash 2.4 from Sterling Data

You may call it activity logging, audit trails, audit logging, change tracking, or something else; whatever name you give to keeping track of changes to your data, SterlingData's LogFlash can do the job.

*Posted Tuesday, October 02, 2001*

## WinInet.DLL: Transferring Files With FTP (Part 2)

In this three part series Matt Grossmith shows how to use the ubiquitous WinInet DLL to transfer files with FTP, delete and rename files on an FTP server, and install callback procedures to monitor file transfer progress.

*Posted Thursday, October 04, 2001*

## PDF for September 30-October 6, 2001

All ClarionMag articles for September 30-October 6, 2001 in PDF format.

*Posted Friday, October 05, 2001*

## WinInet.DLL: Transferring Files With FTP (Part 3)

In this three part series Matt Grossmith shows how to use the ubiquitous WinInet DLL to transfer files with FTP, delete and rename files on an FTP server, and install callback procedures to monitor file transfer progress.

[Clarion Handy Tools Updated](#)

[Icetips Report Wizard Special Offer](#)

[Outlook Express Template Updated](#)

[Icetips Web Site Gets New Look](#)

[HTML Designer Updated](#)

[Clarion 6 To Support Legacy](#)

[ABC Process Progress Bar Replacement](#)

[Fomin Report Builder v.2.8 Released](#)

[SealSoft Free xFunction Library v1.3](#)

[Free Shuffler Template](#)

[TPS.repair Template Beta 3](#)

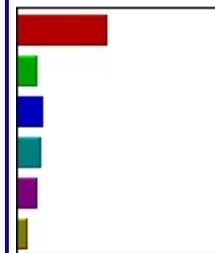
[Clarion 5.506 Released](#)

[ClarionNET 1.1 Released](#)

[Outlook Express](#)

## SURVEY

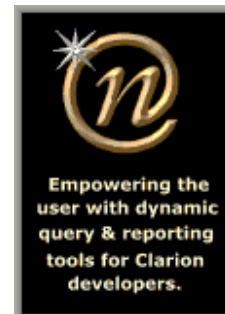
When do you plan to add XML capability to your applications?



■ No Plans (45%)  
 ■ Developing now (10%)  
 ■ In 3 months (13%)  
 ■ In 6 months (13%)  
 ■ In 1 year (10%)  
 ■ More than 1 year away (6%)

229 Total Votes

[Previous Surveys](#)



Nice Touch Solutions, Inc.

*Posted Tuesday, October 09, 2001*

### **Sneak Preview: OLE Classes From PlugWare**

Andy Ireland's new company, Plugware Solutions.com Ltd, will soon be releasing a set of OLE classes that use the Windows API directly, rather than the built-in Clarion OLE functionality.

*Posted Wednesday, October 10, 2001*

### **MySQL: InnoDB Tables And Transactions (Part 1)**

There have been a number of important developments in the MySQL world in the past year, and it's time for an update on how to best use this freely available SQL server with Clarion. This week Dave Harms kicks things off with a discussion of newly-added transaction support and other related features.

*Posted Thursday, October 11, 2001*

### **IRC Transcript: Andy Ireland Discusses COM**

This is an excerpt from the IRC chat on Saturday, October 6, 2001. Andy Ireland, of PlugWare.Com Solutions, Ltd talked about interfaces, COM objects and how to use them in Clarion. Side conversations, parts and joins and other notices are removed for clarity. Also, the conversation is reformatted to read like an interview. You do not need a subscription to read this PDF.

*Posted Monday, October 15, 2001*

### **PDF For October 7-13, 2001**

All Clarion Magazine articles for October 7-13, 2001 in PDF format.

*Posted Tuesday, October 16, 2001*

### **MySQL: InnoDB Tables And Transactions (Part 2)**

In Part 2 of this series Dave Harms tries out MySQL transactions using the InnoDB table type.

*Posted Wednesday, October 17, 2001*

### **Review: CapeSoft File Explorer v2.1**

Information comes packaged in many forms, and for raw database access Clarion is an awesome tool. Quite often, though, you'll want to incorporate other "pre-packaged" bits of information into your

### **Template Goes Gold**

### **Handy Tools FTP Demos Updated**

### **SealSoft Releases xTipOfDay v1.0**

### **Clarion 5.6 To Include New Threading Subsystem**

### **Clarion Links Site**

### **TPS.repair Template In Beta 2**

### **SmartPartner Article Showcases ClarioNET**

### **HandBase Palm & Pocket PC Database ODBC Driver**

### **Duke Application Shell Special Pricing**

### **ABCFree Templates And Tools Updated**

### **CCS Application Conversion Toolset**

### **SimplyClarion.com Announces Developer Support Forums**

### **Lodestar Software Discussion Groups**

### **MySQL 4.0 Alpha Released**

### **xWord Library v1.5 Released**

### **EasyReport 1.00 Released**

### **Gitano Tools Now Available From**

programs. CapeSoft's File Explorer lets you easily add HTML (web pages), PDF (Adobe Acrobat), WAV (audio), AVI (video), Macromedia Flash animations and more to your Clarion applications.

*Posted Friday, October 19, 2001*

### **Weekly PDF for October 14-20, 2001**

All Clarion Magazine articles for October 14-20, 2001 in PDF format.

*Posted Monday, October 22, 2001*

### **Controlling Printers With DevMode (Part 1)**

A customer's request to save printer bin selections sent Jim Gambon off on an exploration of the printer DevMode structure. In this two part series Jim explains how to create a 16/32 bit DevMode that really works, and how to use it to get full control of your printers.

*Posted Wednesday, October 24, 2001*

### **Clarion Developers Conference 2001 Latin America**

The Clarion Developers Conference 2001 Latin America will be held November 8-10 in Buenos Aires, Argentina. All conference proceedings will be in Spanish only. Accordingly, these pages are also in Spanish only!

*Posted Wednesday, October 24, 2001*

### **New Advertising Rates - Special Deals!**

We've changed the way we sell ad space in Clarion Magazine. Until October 30th you can now buy one month of banner advertising for just \$45, with a guaranteed minimum of 2000 views.

*Posted Thursday, October 25, 2001*

### **The Clarion Advisor: Creating A SOAP Client**

Brian Staff shows a snippet of code that demonstrates using Jim Kane's OLE code to create a Simple Object Access Protocol (SOAP) object in Clarion.

*Posted Friday, October 26, 2001*

### **Large Table Performance in MySQL**

The MySQL server has a reputation for speed, yet some developers are noticing huge performance problems with large tables. What's the real story? Dave Harms tests a 300 megabyte, 5.5 million record

[ClarionShop](#)

[For Sale: Real Estate Closing Software](#)

[Screen Prototyper Updated](#)

[AppInit 2.1 Released](#)

[IceTips Wizards](#)

[Clarion 5.506 General Release Coming Soon](#)

[LogFlash 2.4 Released](#)

[AppInfo Template Updated](#)

[xAnalogClock Class v1.0](#)

[ProDomus Update](#)

table and finds out what's really going on.

*Posted Friday, October 26, 2001*

### **Weekly PDF for October 21-27, 2001**

All Clarion Magazine articles for October 21-27, 2001 in PDF format.

*Posted Monday, October 29, 2001*

### **The Clarion Magazine Free Sampler**

The Clarion Magazine Free Sampler is a representative selection of 28 articles published online in past issues of Clarion Magazine, along with the FAQ and subscription information. There's even a PDF version which you can freely distribute!

*Posted Monday, October 29, 2001*

### **Controlling Printers With DevMode (Part 2)**

A customer's request to save printer bin selections sent Jim Gambon off on an exploration of the printer DevMode structure. In this two part series Jim explains how to create a 16/32 bit DevMode that really works, and how to use it to get full control of your printers. Part 2.

*Posted Tuesday, October 30, 2001*

### **Optimizing DLL Loading - Introduction to Rebasing**

If you create DLLs with Clarion, then you'll definitely want to read this series of articles. By default, Clarion 32 bit DLLs load more slowly than they need to, and are not sharable between processes. You can easily fix this, as Carl Barnes shows, thereby reducing load times and memory requirements.

*Posted Wednesday, October 31, 2001*

### **Free Shuffler Template**

New on the free downloads page: Alex Gorbenko's shuffler template. This template works in conjunction with the standard Clarion Browse and allows users to change position of records by dragging-and-dropping. Demo included.

*Posted Wednesday, October 31, 2001*

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

# Clarion MAGAZINE

## The Clarion Magazine Sweepstakes

Published 2001-10-24

Enter the Clarion Magazine Sweepstakes and **you could win** one of the following:

### First Prize (winner to choose one)

- a **Compaq iPAQ** Pocket PC, or
- a **registration** to the **ETC** Clarion Conference in Gatlinburg in May, 2002

### Additional Prizes

- a **two year** Clarion Magazine subscription or renewal, value \$150
- a **one year** Clarion Magazine subscription or renewal, value \$80
- five **six month** Clarion Magazine subscriptions or renewals, value \$40 each
- a copy of **CW Assistant**, valued at \$99
- two copies of the **Clarion Source Search** utility, valued at \$45 each.
- a copy of G-Cal, valued at \$99
- a copy of G-Calc, valued at \$69
- a copy of G-Buddy, valued at \$99

Note: all of the above subscriptions/renewals can also be taken as back issues, in whole or in part

### How To Enter

There are four ways you can enter the Clarion Magazine Sweepstakes:

### Link to this Sweepstakes Page!

Add a link to this page on your web site, and email [dharms@clarionmag.com](mailto:dharms@clarionmag.com) - we'll add an exchange link here.

### Sweeps Sponsors

- [Carl Barnes Computer Consulting](#)
- [Gitano Software](#)

### Sweeps Partners

- [Encourager Software](#)
- [Brady & Associates](#)

#### Search

#### Home

#### COL Archives

#### Advertise

#### Reader Comments

#### Call for Articles

#### Information

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

#### Downloads

[PDFs](#)

[Freebies](#)

[Open Source](#)

#### Site Index



- [Refer a friend](#) to Clarion Magazine (you supply your friend's name and email address, and we send your friend a one-time only email); or
- Take out a new [subscription](#) to Clarion Magazine (for a free sampler [click here](#)); or
- [Renew](#) your existing Clarion Magazine subscription; or
- Mail a handwritten postcard (see the [rules](#) for details).

That's all there is to it! The sweepstakes ends January 15, 2002, so get your entry (or entries) in now! And start canvassing your friends and co-workers - remember, for every person you refer to Clarion Magazine, you get an entry in the sweepstakes. The more entries you have, the better your chances!

## Rules

The official rules are available [here](#).

## Reader Comments

[Add a comment](#)

**[Is it true this contest ended last January 15, 2001?](#)**  
**[Sweepstakes ends January 15, 2002! So much for my...](#)**

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

# Clarion MAGAZINE

[Search](#)

## Review: LogFlash 2.4 from Sterling Data

by **Tom Hebenstreit**

Published 2001-10-02

So the boss storms out of his office one day, and yells, "Who changed the accounting system to give everyone a twenty percent raise?" To a chorus of "Not me", "No way" and "Wouldn't dare", all proclaim saintly innocence (or maybe they blame it on an absent, disliked employee.) The problem is, if the system itself didn't keep track of changes, there really *wouldn't* be any way to find out who actually did make the change.

I've heard it called by a lot of names, including Activity Logging, Audit Trails, Audit Logging and Change Tracking, but it all adds up to the same thing – keeping a record of who did what to which record, and when they did it. Depending on how critical your application data is, logging can provide anything from a simple convenience to the user (rolling back stupid mistakes) up to satisfying legal requirements in cases where data integrity and trackability are paramount (proving due diligence to auditors or the Feds, for example, or rolling back malicious data changes made by a disaffected employee).

Now, back when Clarion 5.5 was announced SoftVelocity mentioned that activity logging was being added to the ABC template set. Cool, I thought, just what I needed. Unfortunately, when I actually got 5.5 and tested the dbAudit class, I found it was very simplistic. It just dumped the information into a text file, and it was up to the developer to do anything with it – more like web server logs than an integrated system. (Hmmm, I wonder if it originated as part of the iBuild web development effort?) In any case, it was wholly inadequate for my needs.

Fortunately, though, anytime there is an obvious gap in the Clarion feature set it seems some brave soul leaps into the breach. In this case the benefactor is Sterling Data, with a product aptly named LogFlash, which claims to satisfy all of your basic activity logging needs. Let's take a closer look at it and see how it stacks up.

### Major Features

What does LogFlash do? Glad you asked:

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader Comments](#)

[Call for Articles](#)

[Information](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



- Logs changes to data records on a field by field basis
- Optionally logs inserts and deletes
- When a record is deleted there is a template option to store the entire record (and child records) in an archive file, allowing them to be 'undeleted' at a later date
- Memos up to 64K are stored in a separate file to keep down the size of the main log file
- Changes to individual fields can be rolled back from the log file browser
- The scope of a rollback can range from a single change to reversing an entire of series of changes, and even lets you limit the rollback to changes made by a specific user
- Option to purge the log for a given date range
- All source code is supplied so that you can control the look and feel of the LogFlash procedures
- The log files are compatible with most file drivers, including SQL (the default for the files is TPS)
- Multi-language support – translations for English, German, Danish, Portuguese and Spanish are included
- Compatible with Clarion 2.0 and up, both ABC and Legacy (some features not supported in 2.0 due to CW limitations)
- Logging can also be used with Edit-In-Place (ABC only, Clarion 5 and up)

All key LogFlash information is stored in a single log file. This includes the date/time of change, user ID of person making the change (assuming you have some way of identifying your users), computer name (32 bit only), file name, field name, old value, new value, procedure used, unique record ID (your system ID for that record), primary field (any user field that can identify a record) and four optional user fields where you can record any other info you think is relevant to that log file entry.

## **Installation**

After purchasing LogFlash, you are sent an email with product downloading instructions. The LogFlash install is pretty compact (under a megabyte) and I had no problems logging in to the download site and getting it copied to my machine.

The install itself was smooth, with options for where to install LogFlash, where to put the templates, and which template versions to install (ABC, Legacy or both). The install correctly found the most current of the four versions of Clarion installed on my machine (5.5, in this case), and automatically filled in all of the prompts with that information. At the end of the process, the installer offered to display the Help file (always a plus in my book).

On the down side, the templates were not registered for me, and the installer didn't set up links to the Help (I guess I've been spoiled by installs that follow the Clarion 3<sup>rd</sup> Party Association ([C3PA](#)) guidelines.) All in all, though, the process was quick and painless.



## Implementation

The LogFlash Help file has a single topic that lists, step by step, the requirements for adding activity logging to your applications. Starting with the basic prerequisite of registering the templates, the Help walks you through importing the log file definitions into your dictionary, importing the LogFlash support procedures into your application, and then adding the LogFlash templates themselves. One global extension is required, and after that you simply add a procedure extension template to any update form where you want logging.

LogFlash is almost completely template driven – embed points are provided, but their use is optional. Since the LogFlash procedures are imported directly into your applications, you have complete control over their look, feel and behavior. I found that I could do just about everything I wanted to do by filling in prompts and customizing the appearance of the various LogFlash procedures.

**Figure 1. The LogFlash form extension template**

Figure 1 shows the template used for adding logging to a form. I have to admit that on the first try I was a bit confused by the terminology on this tab. I assumed that 'Primary Field' meant the primary key on the file, but a bit of research (i.e., reading the Help) enlightened me. As shown above, the Unique Record ID Field is for the primary system-ID type key value, while the 'Primary Field' is what the *user* would consider to be their key value, e.g., a Company Name, Customer ID, Sales Order Number or whatever. The Client ID Field is an additional location for user-oriented record identifiers. For example, this field could be a branch office number

for a company with multiple locations, or the line item number on a sales invoice. These LogFlash prompts could probably be made a bit clearer, but they do get the job done.

To finish off that tab, the 'Warn if length > 75' prompt causes the template to throw a warning during the generation of your code if a logged field is larger than the default log file old value/new value string lengths of 75 characters. In my case, I added the LogFlash extension to 20-plus forms, and it correctly flagged the three that did have strings larger than 75 characters.

The Excluded Fields tab lets you manually exclude fields from being logged. This is useful if you are programmatically updating other file fields not visible to the user and don't want them to show up in the activity logs. A feature new to version 2.4 also lets you automatically exclude fields by adding a user option of 'NOACTLOG' in your data dictionary.

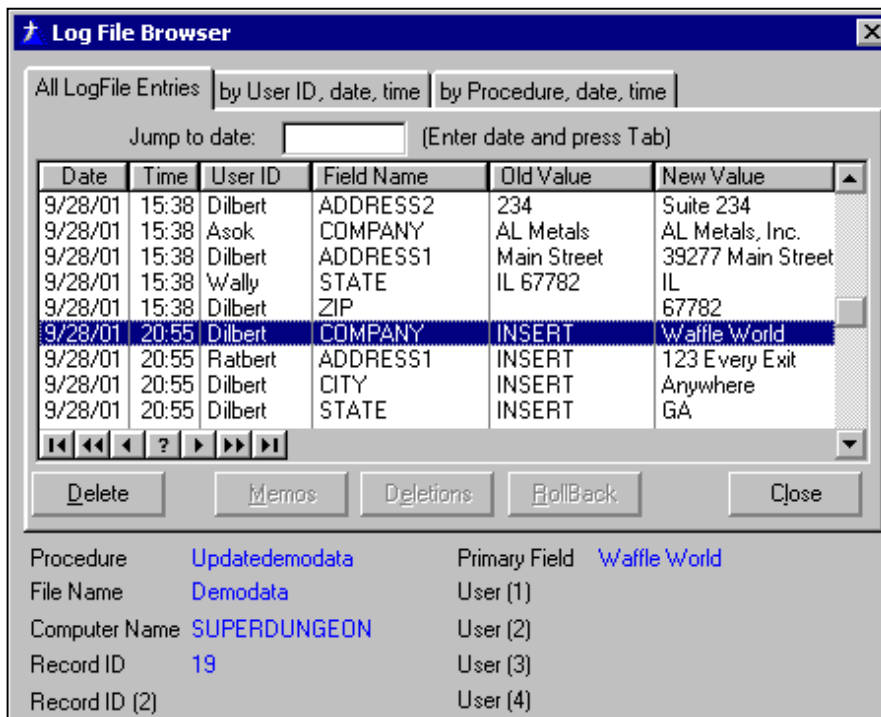
The remaining tabs let you specify whether to log the insertion of new records, how you want to handle deletes (save the record or just log the act of deleting it) and to assign other variables. These include a user ID of the person making the change and up to four other optional fields.

Ok, so you are using LogFlash to create a log of all of the changes to your files (and optionally your inserts and deletes). Now what?

## **The log browser**

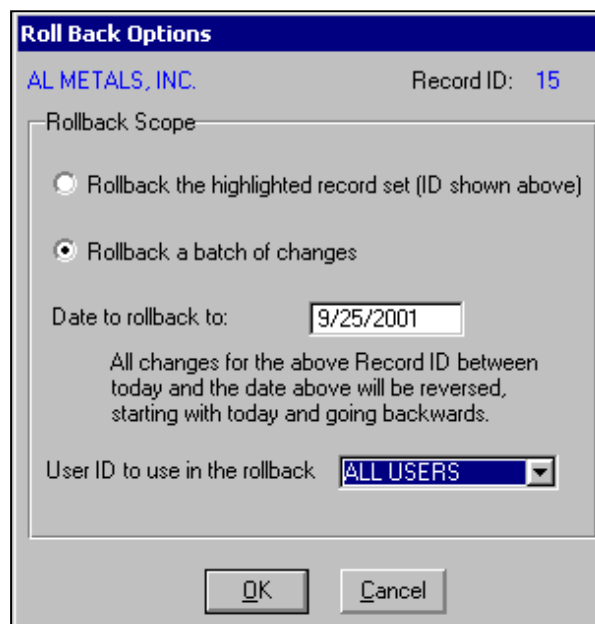
The other side of the equation is filled by a log browser procedure – a place where the user can review a list of the changes that have been made to the data records. LogFlash includes a basic browser that lists all changes to all of the files that you are logging. Note that in Figure 2 you can also see the other fields that LogFlash saves, including the filename, procedure and the name of the computer the user was on when they initiated the action.

### **Figure 2. The default LogFlash activity browser**



The default browser is rather plain but, once again, you can revise it to your heart's content since it is just another procedure in your application. What the default browser really gives you is a framework that already incorporates just about anything you would want to do with your log records. One of the most powerful features that LogFlash has is rollback – the ability to undo one or more changes. Additionally, you can undelete records from here, assuming that you turned on that feature in the templates. The browser also handles enabling and disabling the option buttons based on what you can logically do with a particular record. For example, you can't roll back an insert.

**Figure 3: The default Rollback procedure window**



There is only one other primary function that I haven't yet

covered, and that is keeping control of the size of your activity log file (trust me, these things can grow amazingly fast on a busy system.) LogFlash has that covered as well, providing procedures to enter a date range of records to delete, along with a warning/confirmation procedure and a process to do the actual deletes.

## **Customizing LogFlash**

Because the LogFlash files and procedures are imported into your application, you have tremendous flexibility in tailoring the system to your exact needs. For example, to accommodate a wider range of applications LogFlash uses strings to store the unique record ID. In my test systems, I used auto-incremented longs. I also had fields longer than the LogFlash default of 75 characters. Since the LogFlash 'ActLog' file was just sitting in my dictionary, I was able to easily modify it to match my needs. I swapped data types, extended the old value/new value fields and shrunk some of the default user fields that I didn't need down to one byte. A quick recompile later I had a file that was much more to my liking.

You can also add new fields and keys, if you like. The only thing to be careful of is renaming or removing the default fields and/or keys. It's not that you can't do it; it is just that you will have to make more modifications to the default procedures (and possibly even the templates, if you remove fields they reference.)

In one of my test systems I wanted to see how well LogFlash could be adapted to creating a 'permanent' audit trail, i.e., one that the user could not go back and change. It turned out to be as simple as just hiding the Delete and Rollback buttons on the log browser (I didn't allow purges either, of course).

Finally, I also thought it would be nice to add a History button to my forms that would display the change history for that one record. I didn't need all of the extra features in the standard log browse, so I chose to create a new browse procedure that filtered the ActLog file on the filename and unique record ID fields. Since LogFlash already had a key like that defined, it was a snap.

## **A caveat**

While I was creating my History browse I did find one area that needed some tweaking on my part, at least for my applications. I found that if I added a new record, made changes to it (thus creating log file entries), deleted the record and then added another new record, the new record would inherit the changes from the deleted record. This was logical, in that the new record would have the same auto incremented system ID that the previous new-but-deleted record had. Being logical didn't make it right, though, so I created a simple process to mark all log records for the deleted one as historical and then modified my history browse to exclude those records. I could have just as easily had it delete the prior log records, but this was in the system where I

wanted a permanent record of data changes. I called my process from an embed point (provided by the templates) where LogFlash handles deletions, and then everything worked just the way I wanted it to.

## **Performance**

No matter how you do it, activity logging will have an impact on system speed. The key factors are to balance the tasks that need to be logged versus those that don't, making sure your log file definition is as efficient as possible, and keeping its size under control.

LogFlash tries to keep the record size down by storing only those fields that have changed. This also means, however, that you are going to have more file I/O. For example, if you open a form, change five fields and then save it, you will have not one disk action but six – saving the original record and then inserting five auto incrementing activity log records (one for each old value/new value pair).

That being said, LogFlash performs quite respectably. My tests were done using TPS files, and although I could detect a delay due to the increased disk activity, it was never enough to intrude on my workflow. As the saying goes, your mileage may vary. Just keep in mind that a system that creates activity logs will require a bit more computing power than one that doesn't.

This brings up one final handy feature – the ability to turn logging on and off under program control. Since all of the template generated logging activities are controlled by a single global variable, you can let your users activate/deactivate logging. Just initialize the variable from your INI file at program startup and you are set.

## **Documentation**

Documentation is provided as a single Windows help file. It covers the bases in a logical fashion, beginning with an overview and installation instructions, and then going through the prompts for each template in the order you will most likely use them. An FAQ topic answers common questions, and the file also includes information on contacting Sterling Data.

One improvement I could suggest would be to include the name of the LogFlash translation file in the FAQ section. I was trying to change some of the prompts on the log browse window, but they were being overridden by values from the file. I finally found the name in the installation instructions, but it was not the first place I thought of looking.

Two versions of a demo application are also included, one as standalone and one to illustrate using LogFlash in a multi-app DLL application. The demo is a decent introduction to LogFlash, and

provides an easy way to tinker with various template prompts and view the results in action. I had no problems compiling and using the demo app.

## Technical support

The support from Sterling Data was absolutely stellar. I initially started this review using version 2.3 of the LogFlash templates. After testing it for a few days and digging around in the generated code, I exchanged a series of emails with Sterling Data regarding the implementation of various features along with a few suggestions on my part. The result was that I received version 2.4 the next day! And yes, most of my concerns and suggestions had been incorporated into the new template.







Can't get much better than that, folks!

## Summary

What can I say? I like it.

LogFlash was full-featured, easy to use, easy to adapt and didn't clash with any of the numerous other third-party templates I use in my applications. The generated code is clear and easy to follow, and there are no black boxes to give you grief.

Bottom line: I voted with my own dollars and bought a copy for myself. And that's about the best recommendation I can give to product!

<b>Overall Product Rating:</b>	
Ability to do the task	
Ease of use	
Ease of Installation	
Documentation	
Technical Support	
Black-Box DLLs/LIBs	No

LogFlash 2.4 is \$195 US. More information and a free downloadable demo can be found at the [Sterling Data web site](#).

LogFlash and other Sterling Data products can be purchased at both <http://www.sterlingdata.com> and at <http://www.clarionshop.com>.

## Vendor comments from Sterling Data

*Just a few comments that expand on some of the points raised in the review:*

*Installation - I joined the Clarion 3rd Party Association a few months ago and I have been steadily converting all my templates to use the standard C3PA guidelines. With version 2.4 of LogFlash I will change the install so that it also follows these guidelines.*

*I like the idea for a History option and I will see if I can add this to the next release.*

*I'll give [the Language Translation File] a more prominent position in the help file to make it easier for anyone wanting to change window text.*

---

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

## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

Reborn Free

CLARION  
*online*published by  
CoveComm Inc.

## Clarion MAGAZINE

**WinInet.DLL: Transferring Files With FTP (Part 2)****by Matt Grossmith**

Published 2001-10-04

[Last week](#) I covered some WinInet basics, including converting Visual Basic examples to Clarion, and getting a handle with a call to the WinInet InternetOpen function. Now it's time to use that handle to transfer files between computers using the File Transfer Protocol (FTP).

If you have access to the Internet and a handle to a server connection, all you need are a couple of file names and a function call, and you'll be able to download a file. Simply call the FtpGetFile function:

```
FTPTXFileName = 'avi.txt'
LocalTXFilename = 'avi_from_icetips.txt'
ReturnValue = FTPGetFile(
    InternetConnectionHandle, |
    FTPTXFileName, |
    LocalTXFilename,
    0, |
    0, |
    FTP_TRANSFER_TYPE_UNKNOWN, |
    0)
```

The FTPGetFile API call expects the following parameters:

ConnectionHandle	A handle to an FTP session. Returned from a prior call to InternetConnect.
RemoteFileName	The name of the remote file.
LocalFileName	The name of the local file.
FailIfExists	If this is set to True the download will fail if the local file already exists. For this example I've set this to False (0) so any existing file will be overwritten.

[Search](#)[Home](#)[COL Archives](#)[Advertise](#)[Reader  
Comments](#)[Call for  
Articles](#)[Information](#)[Log In](#)[Membership/](#)[Subscriptions](#)[FAQ](#)[Privacy Policy](#)[Contact Us](#)[Downloads](#)[PDFs](#)[Freebies](#)[Open Source](#)[Site Index](#)



FlagsAndAttribs	The file attributes for the new local file. The value of this parameter can be any combination of the FILE_ATTRIBUTE_* equates. For this example I've set this to be 0 or in other words I have not set any of the files attributes.
Flags	Alters the way the file is downloaded. There are two parts to this value, the first determines the transfer type and the second determines the caching options.
Context	An application defined value that is used for callbacks. I'll cover the callback issue later in this article.

The last Flags attribute can have the following values for FTP transfers:

FTP_TRANSFER_TYPE_ASCII	Transfers the file using FTP's ASCII (Type A) transfer method. Control and formatting information is converted to local equivalents.
FTP_TRANSFER_TYPE_BINARY	Transfers the file using FTP's Image (Type I) transfer method. The file is transferred exactly as it exists with no changes. This is the default transfer method.
FTP_TRANSFER_TYPE_UNKNOWN	Defaults to FTP_TRANSFER_TYPE_BINARY
INTERNET_FLAG_TRANSFER_ASCII	Transfers the file as ASCII.
INTERNET_FLAG_TRANSFER_BINARY	Transfers the file as binary

Caching flags can be any combination of the following, however these are mainly used in HTTP:

INTERNET_FLAG_HYPERLINK	Forces a reload if there was no Expires time and no LastModified time returned from the server when determining whether to reload the item from the network.
-------------------------	--

INTERNET_FLAG_NEED_FILE	Causes a temporary file to be created if the file cannot be cached.
INTERNET_FLAG_RELOAD	Forces a download of the requested file, object, or directory listing from the origin server, not from the cache.
INTERNET_FLAG_RESYNCHRONIZE	Reloads HTTP resources if the resource has been modified since the last time it was downloaded. All FTP and Gopher resources are reloaded.

The `FTPGetFile` function will return `True` if the file downloads successfully, or `False` if there was a problem. If the return value is `False` you can use the error handling function in the downloadable code to determine the exact error.

Uploading is as simple as downloading; you just call the `FtpPutFile` function:

```
FTPTXFileName = 'avi.txt'
LocalTXFilename = 'avi_from_ictips.txt'
ReturnValue = FtpPutFile(
    InternetConnectionHandle, |
    LocalTXFilename, |
    FTPTXFileName, |
    FTP_TRANSFER_TYPE_UNKNOWN, |
    0)
```

The parameters for `FTPputFile` are simpler than the `FTPGetFile` as the `FailIfExists` and `FlagsAndAttributes` parameters are not needed. Also the order of the files is reversed (as you are putting not getting)

## Close the connections

Finally you need to close the connections you've made, in reverse order to their creation, using `InternetCloseHandle`:

```
ReturnValue = InternetCloseHandle|
(InternetConnectionHandle)
```

```
ReturnValue = InternetCloseHandle|
(InternetOpenHandle)
```

`InternetCloseHandle` expects to be passed a valid handle and will return `True` if the handle was closed or `False` if it wasn't. If the return value is `False` `GetLastError` will give the error.

## Putting it all together

Basically all you need to do to download or upload a file via FTP is open an Internet connection with `InternetOpen`, connect with `InternetConnect`, upload or download with `FtpGetFile` or `FtpPutFile`, and then close the connection and free the handle. Listing 3 shows a simple program that does this.

### Listing 3. Putting it all together

```
!Handles
InternetOpenHandle      long
InternetConnectionHandle long
FTPFindFileHandle      long!Variables
ApplicationName         CString('MyApplication')
NullString              CString('')
URL                     CString(261)
UserName                CString(100)
Password                CString(100)
FTPTXFileName          CString(261)
LocalTXFilename        CString(261)
ReturnValue             LONG
!Equates
INTERNET_OPEN_TYPE_DIRECT EQUATE(1)
INTERNET_DEFAULT_FTP_PORT EQUATE(21)
INTERNET_SERVICE_FTP      EQUATE(1)
INTERNET_FLAG_PASSIVE     EQUATE(08000000h)
INTERNET_FLAG_ACTIVE     EQUATE(0)
FTP_TRANSFER_TYPE_UNKNOWN EQUATE(0)
Code
! Initialise
InternetOpenHandle = InternetOpen(
  (ApplicationName, |
  INTERNET_OPEN_TYPE_DIRECT, |
  NullString, |
  NullString, |
  0)
If InternetOpenHandle = 0
  Message('Initialising WinInet failed', |
    'Aborting...', icon:exclamation)
  Return
End
!Connect
URL = 'ftp.icetips.com'
UserName = 'guest'
Password = 'my@emailaddress.com'
InternetConnectionHandle = InternetConnect( |
  InternetOpenHandle, |
  URL, |
  INTERNET_DEFAULT_FTP_PORT, |
  UserName, |
  Password, |
  INTERNET_SERVICE_FTP, |
  INTERNET_FLAG_ACTIVE, 0)
If InternetConnectionHandle = 0
  Message('Connecting to server failed', |
```

```

        'Aborting...',icon:exclamation)
    ReturnValue = InternetCloseHandle(InternetOpenHandle)
    Return
End
!Download
FTPFileName = 'pub/avi.txt'
LocalTXFilename = 'c:\avi_from_icetips.txt'
ReturnValue = FTPGetFile|
    (InternetConnectionHandle, |
    FTPFileName, |
    LocalTXFilename, |
    0,0, |
    FTP_TRANSFER_TYPE_UNKNOWN,0)
If ReturnValue = 0
    Message('The file ' & clip(FTPFileName) & |
        ' could not be downloaded. ', |
        'Error downloading...',icon:exclamation)
Else
    Message('The file ' & clip(FTPFileName) & |
        ' was downloaded to ' & clip(LocalTXFilename) & |
        '.', 'Download complete...',icon:exclamation)
End
!Upload
FTPFileName = |
    'pub/avi_was_downloaded_now_uploaded.txt'
LocalTXFilename = 'c:\avi_from_icetips.txt'
ReturnValue = FtpPutFile|
    (InternetConnectionHandle, |
    LocalTXFilename, |
    FTPFileName, |
    FTP_TRANSFER_TYPE_UNKNOWN,0)
If ReturnValue = 0
    Message('The file ' & clip(FTPFileName) & |
        ' could not be uploaded.', |
        'Error uploading...',icon:exclamation)
Else
    Message('The file ' & clip(LocalTXFilename) & |
        ' was uploaded to ' & clip(FTPFileName) & |
        '.', 'Upload complete...',icon:exclamation)
End
!Disconnect
ReturnValue = InternetCloseHandle(InternetConnectionHandle)
ReturnValue = InternetCloseHandle(InternetOpenHandle)

```

## Adding functionality

The above example shows you how to upload and download files, but it also assumes you already know what files you want. What if you only know what server the files are on, but not which directory they may be in? What you need is a way to look around the server and see what's there. To do this you will use a few more functions from WinInet, namely `FtpGetCurrentDirectory`, `FtpSetCurrentDirectory`, `FtpFindFirstFile` and `InternetFindNextFile`

## Navigating directories

When you connect to an FTP server you normally arrive in the root of the FTP area. With some servers (such as Demon in the UK) you actually arrive in a sub directory of the FTP area that is dedicated to you. Although you can go back up a directory to see the directories owned by other subscribers you can not actually access these other areas.

To find out exactly where you are on the FTP server you can call `FtpGetCurrentDirectory`. This will load one of the parameters you have passed with the directory you are in on the server. The syntax is as follows:

```
FTPCurrentPath      CSTRING(261)
MAX_Path            LONG(260)
CODE
ReturnValue = FtpGetCurrentDirectory( |
    InternetConnectionHandle, |
    FTPCurrentPath, |
    MAX_Path )
```

If the return value is `True` then `FtpGetCurrentDirectory` will have set `FTPCurrentPath` with the FTP directory you are currently in; if the return value is `False` then an error has occurred. You can determine the error by calling `GetLastError`.

The `FTPCurrentPath` is a `CString` of 261 characters passed by reference. I Make this 261 characters because the maximum return value is 260 characters, plus one for the null terminator. The `MAX_Path` parameter is a `Long` set to the maximum number of characters the `FTPCurrentPath` variable can accept, again passed by reference.

To change directory on the FTP server, use the `FtpSetCurrentDirectory` function. This function is similar to the `FtpGetCurrentDirectory`; use it as follows:

```
FTPCurrentPath = clip('pub')
ReturnValue = FtpSetCurrentDirectory( |
    InternetConnectionHandle, |
    FTPCurrentPath)
```

Again a return value of `True` indicates successfully changing directory, a `False` return indicates an error that can be viewed via `GetLastError`.

The only other thing you may need to do with directories is create a directory. Unsurprisingly, you do this with the `FTPCreateDirectory` function:

```
FTPCurrentPath = clip('pub')
ReturnValue = FTPCreateDirectory( |
    InternetConnectionHandle, |
    FTPCurrentPath)
```

As I'm sure you know by now, a return value of `True` indicates successfully creating a directory, while a `False` return indicates an error that can be viewed via `GetLastError`.

Please note that to use any of these functions you must have a handle to a connection on the server.

## Retrieving file lists

Changing directories is fairly easy; getting a list of files is a little trickier. First you get a handle to a file matching the file mask you are looking for. Then you use this handle to repetitively find the next file matching this file mask until there are no files left. There is a special data structure, `WIN32_FIND_DATA`, which handles the file information; it is similar in principle to the structure used in Clarions Directory function:

```

WIN32_FIND_DATA_Def      Group, type
dwFileAttributes         Long
ftCreationTime_dwLowDateTime   Long
ftCreationTime_dwHighDateTime  Long
ftLastAccessTime_dwLowDateTime Long
ftLastAccessTime_dwHighDateTime Long
ftLastWriteTime_dwLowDateTime  Long
ftLastWriteTime_dwHighDateTime Long
nFileSizeHigh           Long
nFileSizeLow            Long
dwReserved0             Long
dwReserved1            Long
cFileName               CString(261)
cAlternate              CString(15)
                        End

```

I've declared `WIN32_FIND_DATA_Def` as a `TYPED` group, which means you'll need to create an instance of this group before you can use it. You'll then pass that group to the function, and when the function completes the group will contain data about the found file. You can then extract data for storage elsewhere (i.e. a queue, for display). Here's a group declaration and a queue:

```

Win32_FindData Group(Win32_Find_Data_Def),pre(LCLW32FD)
                        End
FTPFindHandle Long
SearchFor      CString(261)
RemoteFileQ    Queue,pre(RFQ)
Filename       CString(261)
Attribs        long
                        End

```

The first thing you'll need to do is get a handle to the first file matching your file mask; if you want to list all files on the server the file mask would be `*.*`:

```

SearchFor = '*.*'
FTPFindHandle = FtpFindFirstFile( |
InternetConnectionHandle, |

```

```

SearchFor, |
Win32_FindData, 0, 0)
if FTPFindHandle
  RFQ:Filename = LCLW32FD:cFileName
  RFQ:Attribs = LCLW32FD:dwFileAttributes
  Add(RemoteFileQ)
  Loop
    ReturnValue = InternetFindNextFile( |
      FTPFindHandle, |
      LCL:Win32_FindData)
    If ReturnValue = 0 then break.
    RFQ:Filename = LCLW32FD:cFileName
    RFQ:Attribs = LCLW32FD:dwFileAttributes
    Add(RemoteFileQ)
  End
End
InternetCloseHandle(FTPFindHandle)

```

**FtpFindFirstFile** expects to be passed the following parameters:

ConnectionHandle	A handle to an FTP session. Returned from a prior call to <code>InternetConnect</code> .
FileMask	A string that contains a valid path or filename. It can include wildcards but no blank spaces. If blank then the first file in the server directory is found.
FindDataStructure	The structure to receive the file information.
Flags	Alters the way the file is found. Values can be any combination of:  <code>INTERNET_FLAG_HYPERLINK</code> <code>INTERNET_FLAG_NEED_FILE</code> <code>INTERNET_FLAG_NO_CACHE_WRITE</code> <code>INTERNET_FLAG_RELOAD</code> <code>INTERNET_FLAG_RESYNCHRONIZE</code>
Context	An application defined value that is used for callbacks. I'll cover the callback issue later in this article.

**FtpFindFirstFile** will return a handle to the file if successful or zero for an error. The error can be found by calling `GetLastError`. If successful the handle is used in subsequent calls to `InternetFindNextFile`.

**InternetFindNextFile** expects to be passed the following parameters:

FileHandle	A handle to a file. Returned form a prior call to FtpFindFirstFile.
FindDataStructure	The structure to receive the file information.

InternetFindNextFile will return True if a file was found or False if no file matched the search criteria. Once there are no files left you break out of the loop. Remember to close the file handle! If you do not close this handle then subsequent call to FtpFindFirstFile and InternetFindNextFile will not work correctly.

I don't have a complete list of the file attributes for files sitting on a server but be aware they are *not* the same as local file attributes! I do know however that if the attribute is 16 then the file is actually a directory.

## Deleting or renaming files and directories

Now that you can find files and directories, download and upload, about the only thing left to do is rename or delete files. This is pretty simple really; the FTPRenameFile function works on both files and directories, and the FTPDeleteFile and FTPRemoveDirectory work on files and directories respectively. Go ahead and rename the file you uploaded earlier:

```

FTPTXFileName = |
    'pub/avi_was_downloaded_now_uploaded.txt '
FTPTXNewFileName = |
    'pub/avi_was_downloaded_now_uploaded_and_renamed.txt '
ReturnValue = FtpRenameFile(|
    InternetConnectionHandle, |
    FTPTXFileName, |
    FTPTXNewFileName)
! the above code renames a file,
! the following code renames a directory
FTPTXFileName = 'pub'
FTPTXNewFileName = 'old_pub'
ReturnValue = FtpRenameFile(
    InternetConnectionHandle, |
    FTPTXFileName, |
    FTPTXNewFileName)

```

Deleting a file is just as easy:

```

FTPTXFileName = |
    'pub/avi_was_downloaded_now_uploaded_and_renamed.txt '
ReturnValue = FtpDeleteFile(|
    InternetConnectionHandle, |
    FTPTXFileName)
! the above code deletes a file,
! the following code deletes a directory
FTPTXFileName = 'pub'
ReturnValue = FTPRemoveDirectory(|
    InternetConnectionHandle, |

```



FTPTXFileName )

That covers the basics of uploading and downloading files using FTP and WinInet. But there's more! Some WinInet functions allow the use of callbacks, which are procedures that you create and WinInet calls. You can use these to inform your users of the progress of the file transfer. I'll cover these next week.

[Download the source](#)

## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

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

# Clarion MAGAZINE

## WinInet.DLL: Transferring Files With FTP (Part 3)

**by Matt Grossmith**

Published 2001-10-09

As I mentioned in Parts 1 and 2, some of the functions in the Wininet.DLL allow the use of callbacks. For those of you who don't know, a callback is a way of allowing a function or procedure to periodically "call back" to your application to allow for some other thing to happen. For instance, if you are using an API function that will take a long time to complete, that function may allow you to specify a callback that would increment a counter. You pass the API function the address of a properly prototyped Clarion function, and the API function calls the Clarion procedure, which displays information to the user.

The WinInet functions that I'm using which support callbacks are FTPGetFile, FTPPutFile and FTPFindFirstFile. WinInet allows you to create a single callback function that will handle all callbacks, or different callback functions for each Internet handle you create. I will be using just one callback function to handle all the callbacks in this article. As the same function is used for each callback there needs to be some means of identifying what is being called back. This is where the "context" value comes in. The MSDN describes the context value as a "pointer to an unsigned long integer value that contains the application-defined value that associates this search with any application data." Just think of the context value as a number that identifies what is calling the callback function. To use callbacks I will need to add another function to the WinInet declarations:

```
module('wininet')
...
  InternetSetStatusCallback(long,long), |
  long,raw,pascal,name(|
  'InternetSetStatusCallbackA')
end
```

The `InternetSetStatusCallback` function initializes the use of callbacks and tells WinInet the address of the function you define that will deal with callbacks for the handle specified. You must also specify that context I mentioned in the calls to the functions that support callbacks. So if you created a Clarion procedure called

[Search](#)[Home](#)[COL Archives](#)[Advertise](#)[Reader  
Comments](#)[Call for  
Articles](#)[Information](#)[Log In](#)[Membership/](#)[Subscriptions](#)[FAQ](#)[Privacy Policy](#)[Contact Us](#)[Downloads](#)[PDFs](#)[Freebies](#)[Open Source](#)[Site Index](#)etc-III  
Clarion  
Conference  
Sponsor

**CallbackHandler for your current connection**  
(InternetConnectionHandle in the above examples) the code would be as follows:

```
AddressOfCallbackHandler    Long
...
Code
...
AddressOfCallbackHandler = address(CallbackHandler)
ReturnValue = InternetSetStatusCallback( |
    InternetConnectionHandle, |
    AddressOfCallbackHandler)
...
```

InternetSetStatusCallback will return the address of the previously defined callback handler if there was one, NULL if there was not a previously defined callback handler or INTERNET\_INVALID\_STATUS\_CALLBACK if the callback handler is invalid.

The function that handles callbacks, CallbackHandler in this case, must match the prototype the called function is expecting. In the case of WinInet this prototype is:

```
CallbackHandler(long, long, long, long, long), raw, pascal
```

The five longs are as follows:

ConnectionHandle	A handle to an FTP session
Context	An application defined value that is used for callbacks
InternetStatus	A long that contains the status code that indicates why the callback function is being called.
StatusInformation	A pointer to a structure that contains information pertinent to this call to the callback function
StatusInformationLength	A long containing the length of the structure

Valid values for the InternetStatus code are:

INTERNET_STATUS_⬅ _CLOSING_CONNECTION	Closing the connection to the server. The StatusInformation pointer is NULL.

INTERNET_STATUS↵ _CONNECTED_TO_SERVER	Successfully connected to the socket address (SOCKADDR) pointed to by StatusInformation.
INTERNET_STATUS↵ _CONNECTING_TO_SERVER	Connecting to the socket address (SOCKADDR) pointed to by StatusInformation.
INTERNET_STATUS↵ _CONNECTION_CLOSED	Successfully closed the connection to the server. The StatusInformation pointer is NULL.
INTERNET_STATUS↵ _CTL_RESPONSE_RECEIVED	Not implemented.
INTERNET_STATUS↵ _DETECTING_PROXY	Notifies the client application that a proxy has been detected.
INTERNET_STATUS↵ _HANDLE_CLOSING	This handle value has been terminated.
INTERNET_STATUS↵ _HANDLE_CREATED	Used by InternetConnect to indicate it has created the new handle. This lets the application call InternetCloseHandle from another thread, if the connect is taking too long. The StatusInformation pointer contains the address of an INTERNET_ASYNC_RESULT structure.
INTERNET_STATUS↵ _INTERMEDIATE_RESPONSE	Received an intermediate (100 level) status code message from the server.
INTERNET_STATUS↵ _NAME_RESOLVED	Successfully found the IP address of the name contained in StatusInformation.
INTERNET_STATUS↵ _PREFETCH	Not implemented.
INTERNET_STATUS↵ _RECEIVING_RESPONSE	Waiting for the server to respond to a request. The StatusInformation pointer is NULL.

INTERNET_STATUS← _REDIRECT	An HTTP request is about to automatically redirect the request. The StatusInformation pointer points to the new URL. At this point, the application can read any data returned by the server with the redirect response and can query the response headers. It can also cancel the operation by closing the handle. This callback is not made if the original request specified INTERNET_FLAG_NO← _AUTO_REDIRECT.
INTERNET_STATUS← _REQUEST_COMPLETE	An asynchronous operation has been completed. The StatusInformation pointer contains the address of an INTERNET_ASYNC_RESULT structure.
INTERNET_STATUS← _REQUEST_SENT	Successfully sent the information request to the server. The StatusInformation pointer points to a DWORD containing the number of bytes sent.
INTERNET_STATUS← _RESOLVING_NAME	Looking up the IP address of the name contained in StatusInformation.
INTERNET_STATUS← _RESPONSE_RECEIVED	Successfully received a response from the server. The StatusInformation pointer points to a DWORD containing the number of bytes received.
INTERNET_STATUS← _SENDING_REQUEST	Sending the information request to the server. The StatusInformation pointer is NULL.
INTERNET_STATUS← _STATE_CHANGE	<p>Moved between a secure (HTTPS) and an unsecure (HTTP) site. This can be one of the following values:</p> <p>INTERNET_STATE← _CONNECTED - Connected state (mutually exclusive with disconnected state).</p> <p>INTERNET_STATE← _DISCONNECTED - Disconnected state. No network connection could be established.</p>

INTERNET\_STATE←  
 \_DISCONNECTED\_BY\_USER -  
**Disconnected by user request.**

INTERNET\_STATE\_IDLE - No  
**network requests are being made  
 by the Microsoft® Win32®  
 Internet functions.**

INTERNET\_STATE\_BUSY - Network  
**requests are being made by the  
 Win32 Internet functions.**

INTERNET\_STATUS←  
 \_USER\_INPUT\_REQUIRED - The  
**request requires user input to be  
 completed.**

The simplest way of using the callback handler to show the user that something is happening is to use a progress bar on the main frame and update it every time the callback handler is called:

```
CallbackHandler  PROCEDURE(ConnectionHandle, |
                    Context, |
                    Status, |
                    Info, |
                    Len)

CODE
! ProgressBarFEQ is the field equate
! of a progress bar on the main
! frame with a range of 0 - 100
ProgressBarFEQ{PROP:progress} = |
  ProgressBarFEQ{PROP:progress} + 1
If ProgressBarFEQ{PROP:progress} >= 100
  ProgressBarFEQ{PROP:progress} = 0
End
```

This code will make the progress bar move whenever the callback handler is called, no matter which API function makes the call. It doesn't give the user much information but at least it indicates the system has not hung while a large download is happening.

If you only want to update the progress bar for downloads then you will need to check for a specific context in the callback handler. The simplest way of doing this is to create an equate for "Downloading a file" and use this equate in the callback and FTPGetFile functions:

```
DownloadingAFile  equate(1)
...
Code
...
InternetConnectionHandle = InternetConnect( |
  InternetOpenHandle, |
```

```

URL, |
INTERNET_DEFAULT_FTP_PORT, |
UserName, |
Password, |
INTERNET_SERVICE_FTP, |
INTERNET_FLAG_ACTIVE, |
0)
ReturnValue = InternetSetStatusCallback(|
    InternetConnectionHandle, |
    AddressOfCallbackHandler)
...
ReturnValue = FTPGetFile(|
    InternetConnectionHandle, |
    FTPTXFileName, |
    LocalTXFilename, |
    0,0, |
    FTP_TRANSFER_TYPE_UNKNOWN, |
    DownloadingAFile)
...

CallbackHandler      PROCEDURE(ConnectionHandle,
                                Context,
                                Status,
                                Info,
                                Len)

CODE
! ProgressBarFEQ is the field equate
! of a progress bar on the main frame
! with a range of 0 - 100
if Context = DownloadingAFile
    ProgressBarFEQ{PROP:progress} = |
        ProgressBarFEQ{PROP:progress} + 1
    if ProgressBarFEQ{PROP:progress} >= 100
        ProgressBarFEQ{PROP:progress} = 0
    end
end
end

```

There is a fuller example of callback handling in the downloadable code at the end of this article.

## Summary

That just about covers the use of the basic WinInet functions. If you've followed the above article you can now add FTP functionality to any of your applications. You'll also find that many of the other WinInet functions are similarly easy to use. You can create applications that dial up an Internet connection, request and retrieve web pages, deal with proxies, and much more. And to give your users a sense of what's happening, you can pass callback procedures to some WinInet API calls.

[Download the source](#)

## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.



# Clarion MAGAZINE

## Clarion News

### [Clarion Handy Tools Updated](#)

The Clarion Handy Tools build O7A-1 is now available. As of this release Clarion Handy Tools provide 97 templates and 70 classes with functionality including: browse enhancement; language querying for browses, processes, reports and web pages; email; FTP; HTTP; and much more. Highlighted in this build: HTTP browser server technology: you can serve your data to the Internet or an intranet without using server software like IIS or Apache, and without the Clarion Application Broker. Two demos available.

*Posted Tuesday, October 30, 2001*

---

### [Icetips Report Wizard Special Offer](#)

Icetips Software has released the next in its line of Wizards for the Clarion developers. Report Wizard includes a Wizard Editor that manages report layouts; just select a layout when you run the wizard. Report Wizard supports Clarion 4, Clarion 5, Clarion 5.5, ABC and Legacy and CPCS. Features include: define your own variables in the layout and then populate them on the report; add your own embed code to the layout; create as many details as you want for each layout, complete with font settings; positions and sizes are automatically calculated to fit the detail area(s) on the report. Until October 31, 2001 the single user version is US\$ 99.00, \$50 off the regular price of US\$149.00.

*Posted Monday, October 29, 2001*

---

### [Outlook Express Template Updated](#)

ExpressFlash now links messages into your "people files" by matching From and To email addresses and then linking the messages file to employee and customer records. You can quickly see all the emails sent to a particular customer account or which customers an employee has been writing to. New demo available.

*Posted Monday, October 29, 2001*

---

### **Icetips Web Site Gets New Look**

Arnor and Sue have given the Icetips web site a complete redesign, incorporating the Icetips News Network which provides Clarion developers with the latest news about what's happening in the Clarion community. Icetips has been around since June 1995 and was one of the first web sites that was dedicated to Clarion developers. It served as a central place for developers to seek help and download files and all kinds of thing.

*Posted Monday, October 29, 2001*

---

### **HTML Designer Updated**

The latest release of HTML Designer has the following features: import help file information from your Clarion application; import HTML files in the help directory in the HTML designer; list of recently used projects; create HTML help file as RTF; create and use image maps in the HTML files; register HTML designer to obtain unique serial number. Documentation is being created. The beta price is the same, but the Gold release will now be \$124, not the earlier announced \$99.

*Posted Wednesday, October 24, 2001*

---

### **Clarion 6 To Support Legacy**

Bob Zaunere has stated that the next major release of Clarion, version 6, will continue to support the legacy template chain.

*Posted Wednesday, October 24, 2001*

---

### **ABC Process Progress Bar Replacement**

JaDu Technologies has released two control templates which are drop-in replacements for ABC process or report progress bars. These progress bars feature nine built-in styles.

*Posted Wednesday, October 24, 2001*

---

### **Fomin Report Builder v.2.8 Released**

Fomin Report Builder v.2.8 is now available. This release is compatible with ClarionNET 1.1 and Clarion 5.5F, and includes various fixes and enhancements.

*Posted Wednesday, October 24, 2001*

---

### **SealSoft Free xFunction Library v1.3**

SealSoft's free xFunction library version 1.3 includes the following new date functions: xIsLastDayOfMonth, xLastDayOfMonth, xDayOfYear, xDaysLeftInYear. New install and demo available for Clarion 5 and 5.5.

*Posted Wednesday, October 24, 2001*

---

### **Free Shuffler Template**

A new free template, contributed by Alexander Gorbenko, is available in the Clarion Central download section. This Legacy/ABC template allows your customers to change position of records in a browse box by dragging-and-dropping. Includes demo with source.

*Posted Wednesday, October 24, 2001*

---

### **TPS.repair Template Beta 3**

The TPS.repair Template beta 3 is now available. This release supports local linking, a new "single record copy" recovery feature, a new audit logfile extension template to verify that all records could be recovered, and a handful of other new features and fixes. The template set supports Clarion version 5.5 and is available free of charge during the beta.

*Posted Wednesday, October 24, 2001*

---

### **Clarion 5.506 Released**

The latest update to Clarion 5.5 is available for download. This release includes a number of bug fixes and some new features, such as: NextTabStop/PrevTabStop properties; PROP:Boxed for text controls; RECORDs are now permitted structure declarations, similar to GROUPs; MSSQL/ODBC drivers now support NCHAR and NVARCHAR data types; DLL entry break point in debugger.

*Posted Thursday, October 18, 2001*

---

### **ClarionNET 1.1 Released**

SoftVelocity has released the first update to ClarionNET. The 1.1 version contains some significant fixes, and includes the most asked-for new features.

*Posted Thursday, October 18, 2001*

---

### **Outlook Express Template Goes Gold**

Sterling Data has released ExpressFlash 1.0. ExpressFlash lets you import messages from Outlook Express into your app's database. Possibilities include automated order processing, knowledge bases, etc. You can specify a starting date & time for each import run so you don't get any overlap between batches of emails converted. There is also an option to convert Outlook Express mailboxes to "raw" text data - which you can then parse out as you want. Demo available.

*Posted Thursday, October 18, 2001*

---

### **Handy Tools FTP Demos Updated**

The Handy Tools downloadable FTP demos have been updated to use a new ftp site. The demos called FTPWizard and FTPEngine can be configured to use your own FTP sites as well. The demo source is included with the FTP tools purchase.

*Posted Tuesday, October 16, 2001*

---

### **SealSoft Releases xTipOfDay v1.0**

SealSoft has released xTipOfDay, a library with extension template that adds a "Tip Of Day" to any Application. xTipOfDay sells for \$39. Demo available.

*Posted Tuesday, October 16, 2001*

---

### **Clarion 5.6 To Include New Threading Subsystem**

SoftVelocity's Bob Brooker has confirmed that Clarion 5.6 will include a plug-in replacement for the threading subsystem. As this is a replacement, no changes to existing applications should be needed.

*Posted Tuesday, October 16, 2001*

---

### **[Clarion Links Site](#)**

Mark Geisinger has put together a dmoz.org-style site for Clarion tools and products. This site will include banner ads for any interested third party developers.

*Posted Tuesday, October 16, 2001*

---

### **[TPS.repair Template In Beta 2](#)**

Jens Eden's Informatik.Consulting has joined the Clarion 3rd Party Community; Informatik's first template set, the TPS.repair Template, has just gone into public beta. TPS.repair implements file maintenance and repair functionality for files that use SoftVelocity's TopSpeed file format (.TPS files). The Application Wizard Utility generates a standalone, ready to deploy, end user file repair tool, based on a data dictionary, in less than a minute. The generated file repair tool is full featured including compressed backup of original file and error log generation for diagnosis. The templates support Clarion versions 4.0 to 5.5 and are available free of charge at while in beta. The templates will sell on ClarionShop for \$50 after release, which is scheduled for Nov. 1, 2001.

*Posted Tuesday, October 16, 2001*

---

### **[SmartPartner Article Showcases ClarioNET](#)**

Michael Lawson's submission to the SmartPartner RFP showdown is now available online. This submission features a ClarioNET solution to a realty company's database consolidation requirements.

*Posted Tuesday, October 16, 2001*

---

### **[HanDBase Palm & Pocket PC Database ODBC Driver](#)**

Roy Brubaker reports that the new HanDBase ODBC driver works with Clarion 5.5. This driver makes it easy for Clarion applications to work with handheld HanDBase databases.

*Posted Tuesday, October 16, 2001*

---

### **[Duke Application Shell Special Pricing](#)**

Until December 31, 2001 Gerald W Duke and Company is offering the follow specials: any Application Shell for \$99.00; the Enterprise Edition for \$1199.00; the Popular Edition for \$749.00; Office Plus w/Payroll for \$299.00.

*Posted Tuesday, October 16, 2001*

---

### **[ABCFree Templates And Tools Updated](#)**

A new version of ABCFree Templates and Tools is now available. This set includes 24 code templates, 10 utility templates, 150 extension templates, 27 control templates, 27 classes, and 15 examples and tools. This release includes bug fixes and new features.

*Posted Tuesday, October 16, 2001*

---

### **[CCS Application Conversion Toolset](#)**

Cowboy Computing Solutions/DBWired has a toolset available to facilitate the following application migrations: Legacy to SQL, Legacy to ABC, and ABC to SQL.

*Posted Tuesday, October 16, 2001*

---

### **[SimplyClarion.com Announces Developer Support Forums](#)**

SimplyClarion.com Discussions is a community resource for third party developers. Public areas of the site are open for anyone to browse and participate in. Participation (posting messages) requires registration, which is simply a process to supply a user name and email address. Discussions generates a password that is emailed to the registrant, thus verifying a working email. Private forums are available.

*Posted Tuesday, October 16, 2001*

---

### **[Lodestar Software Discussion Groups](#)**

Lodestar Software has opened RPM/AFE discussion groups at SimplyClarion. These groups are free to anyone who registers with SimplyClarion.

*Posted Tuesday, October 16, 2001*

---

### **[MySQL 4.0 Alpha Released](#)**

The MySQL 4.0 alpha is now available for download at the MySQL website. The new version is intended as a platform for building mission critical, heavy load database solutions. Although the current release is an alpha version, the development team expects to release a beta version after rapid initial testing and will also add a number of new features in the weeks to come. Features now available include: an embedded MySQL server library, making it easier to embed the MySQL server in third party software and solutions; SSL connections; enhanced speed; enhanced compatibility with the SQL standard. Features coming soon include: changing startup options without taking down the server; fail safe replication; online hot backup with low performance penalty.

*Posted Tuesday, October 16, 2001*

---

### **[xWord Library v1.5 Released](#)**

SealSoft's xWord Library 1.5 adds numerous new methods to the xWord class for additional flexibility in working with Word documents. A new install and demo are available.

*Posted Tuesday, October 16, 2001*

---

### **[EasyReport 1.00 Released](#)**

Ingasoftplus has released EasyReport version 1.00. Features include greenbar effect and automatic adjustment of text/line/box controls. Class and templates included; supports Clarion 5/5.5, ABC/Legacy, 32 bit only.

*Posted Friday, October 05, 2001*

---

### **[Gitano Tools Now Available From ClarionShop](#)**

ClarionShop has added the complete line of Gitano Software products to its online store.

*Posted Friday, October 05, 2001*

---

### **[For Sale: Real Estate Closing Software](#)**

Ed Schneider has a Real Estate closing package for sale. This software sells to legal and financial markets, specifically firms that represent lenders in the closing process. A complete closing package can be produced in minutes rather than hours. The HUD,

Closing Statement, Note, Mortgage, Affidavits and virtually any closing documents are produced in a fraction of the time required using conventional methods. Write checks and produce client ledgers. Management reports for closing statistics and a Daily Diary to help you manage the closing process are included in Quick Close. Over 1,300 fields can be merged into and edited in Word Perfect. Over 300 predefined forms. Developed in Clarion 4. 40 plus current users (all small to large law firms in New York State).

*Posted Friday, October 05, 2001*

---

### **Screen Prototyper Updated**

The Riebens Systems Screen Prototyper has been updated. The interface was completely re-designed and you can now have your Tools toolbox either floating or docked to the application frame. It is also possible to change the size of the Toolbox. The previewer for the screens will now automatically size to the screen size defined during the prototyping phase. The Screen Reporter have the option of printing all the screen details or switching off the notes and/or the screen controls. The prototyper will run in Demo mode for 30 days and also allows the user to purchase and clear the demo mode. Purchased products will allow developers to check for updates on the web and download and install the new releases. On the web page follow the link under PRODUCTS and then CLARION 3RD PARTY PRODUCTS and then DETAILS. For this release, creating controls from the dictionary has been disabled due to a bug.

*Posted Friday, October 05, 2001*

---

### **AppInit 2.1 Released**

The Sylkie Enterprise has released AppInit 2.1. Features include: automatic build number generation at each compile; automatic generation of Splash Screen and About box; Email and Homepage links on the About box are "hotfields" that are clickable links; string constants in the splash and about screens are translatable into any local language; version stamping of EXE and DLL with the generated version information; DLL versions can be checked against a master list in the main exe for version level compliance; unnecessary ABC classes can be excluded when making DLL modules; an extension template is provided to save and restore global values to an INI file at application start and end; a procedure level extension template is provided to save and



restore procedure level information to an INI file; application can be limited to a single instance; Windows shutdown request can be responded to by shutting application down; version number can be applied to application main caption; copyright years have the option of using the internal date stamp or of using manually entered year; build date can be set to display either the current date, or the date the application was last changed; a system-wide icon can be set. AppInit sells for \$39.

*Posted Thursday, October 04, 2001*

---

### **IceTips Wizards**

The Icetips Wizards are a combination of an ABC Wizard template set and a Wizard Editor. The Wizard Editor allows the developer to set up the look and feel of the windows and reports that the Wizard templates will generate. The Wizard templates also allow the developer for example to specify what fields should be populated in a browse, and in what order; this also applies to reports. The Wizard Editor saves the setup to files, configuration files or schemes; there is no limit to the number of schemes that can be created. When the Wizard template is used, the developer selects what scheme file should be used and the procedures will be created according to that scheme. Demo available. Price is \$99 through September 30, 2001, \$149 thereafter.

*Posted Wednesday, October 03, 2001*

---

### **Clarion 5.506 General Release Coming Soon**

The Clarion 5.506 patch image has been updated and posted for third party compatibility testing. General release is scheduled for later this week.

*Posted Wednesday, October 03, 2001*

---

### **LogFlash 2.4 Released**

New features in LogFlash 2.4: new template prompts for user defined fields; use of NOACTLOG in dictionary field user options to exclude fields from logging on an app-wide basis; template option to convert filenames etc. to upper case; installation program conforms to the Clarion 3rd Party Developers Association guidelines. Demo available.

*Posted Wednesday, October 03, 2001*

---

### **AppInfo Template Updated**

The free AppInfo template has been updated. This freeware template will give you some information about your application, including templates in use, icons in use, wav files in use, dictionary info, etc.

*Posted Wednesday, October 03, 2001*

---

### **xAnalogClock Class v1.0**

New from SealSoft: xAnalogClock class. Features include: time offset from clocks on your computer; clock wallpaper image; text and date or time on clock; numerous options and flexible setup. Demo available. Sold separately from xDigitalClock, or in xClockPack.

*Posted Tuesday, October 02, 2001*


---

### **ProDomus Update**

A number of Prodomus products have received recent updates. The translation dictionary synchronization feature has been modified to provide the user with a list of any differences between translation dictionary files and the current translation file. The dictionary or the translation file may be selectively modified after this procedure is run. The Source Manager and Extraction Utility has been updated to handle the new C55 default cell tool tips. Several other usability enhancements have also been made. A translation dictionary containing a collection of translations from English to several languages is now available to users of the translation utilities and those who have contributed to this file. This contains over 23,000 translations for German, French, Spanish and several other languages. And Translator Plus, PD 1-Touch Date Tools, and PD Lookups have all been modified to support the new ClarioNET thin client Clarion add-on now available from SoftVelocity.

*Posted Monday, October 01, 2001*

---



Copyright © 1999-2001 by CoveComm Inc. All Rights Reserved. Reproduction in any form without the expresswritten 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](http://www.clarionmag.com), email [covecomm@mbnet.mb.ca](mailto:covecomm@mbnet.mb.ca).

# Clarion MAGAZINE

## Sneak Preview: OLE Classes From PlugWare

by **Dave Harms**

Published 2001-10-10

Andy Ireland is well-known to many Clarion developers for his love of low-level coding (and a few other things, but that's another story). Andy's new company, Plugware Solutions.com Ltd, will soon be releasing a set of OLE classes that use the Windows API directly, rather than the built-in Clarion OLE functionality.

Andy's demonstrator app is a Clarion OLE web browser container, as shown in Figure 1. This is just an early example of what the OLE classes can do, and so is missing a few of the niceties such as a URL entry field, but all I needed to do to navigate with the browser was click on links or drag another URL onto the browser.





**Figure 1. The Plugware web browser**

Listing 1 shows the code required create the browser. The code to create the OLE container is in the CreateBrowser method, but even here the code is quite simple; all the heavy lifting is done by the OLE classes.

```

program

    include('WebBrowser.inc'),once
    include('olereg.inc'),once

    map
        CreateBrowser
    end

OleWindow WINDOW('OleWindow'),AT(, ,205,179),|
    SYSTEM,GRAY,RESIZE,IMM
    END

OleIniter          COLEIniter
WebBrwsr           WebBrowser
CtrlSite           &COLECtrlSite
CtrlContainer      COLECtrlContainer
fBrowserCreated    bool
ResizeCount        long

    code
        if OleIniter.IsInitialised()
            open(OleWindow)
            accept
                case event()
                    of event:openwindow
                        CreateBrowser
                    of event:sized
                        if fBrowserCreated and ResizeCount > 0
                            CtrlContainer.Resize().
                            ResizeCount += 1
                        end
                    end
                end
            end
        end
        if ~CtrlSite &= null
            dispose(CtrlSite).

CreateBrowser procedure

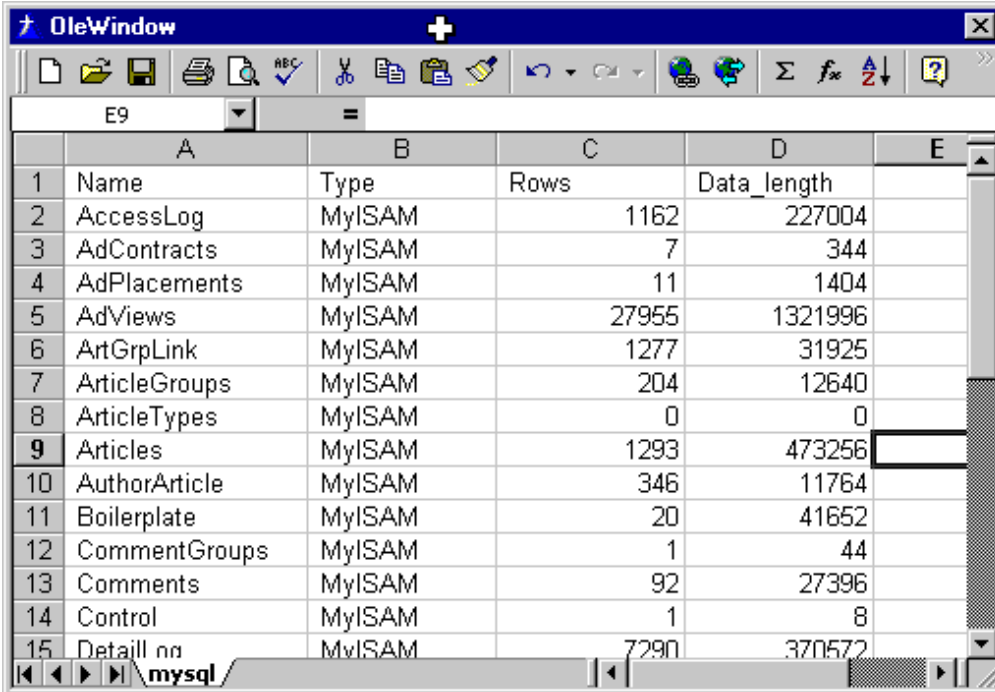
rcWindow          like(_RECT_)
szURL              cstring('www.microsoft.com')
hr                HRESULT,auto
  
```

```

code
  GetClientRect(OleWindow{prop:clienthandle} |
    , rcWindow)
  if CtrlContainer.Init(|
    OleWindow{prop:handle}, |
    OleWindow{prop:clienthandle})
    CtrlSite &= new COLECtrlSite
    CtrlSite.Init(CtrlContainer, |
      OleWindow{prop:clienthandle}, rcWindow)
    hr = WebBrwsr.Init(CtrlSite)
    if hr = S_OK
      hr = WebBrwsr.Navigate(szURL)
      fBrowserCreated = true
    end
  end
end

```

The application is more than just a browser, however. Drag an Excel spreadsheet onto the browser window, and you'll have a spreadsheet container, as shown in Figure 2.



The screenshot shows a window titled 'OleWindow' containing an Excel spreadsheet. The spreadsheet has a toolbar at the top and a data table below. The data table has 15 rows and 5 columns. The columns are labeled A through E. The rows contain data for various categories, including Name, Type, Rows, and Data\_length.

	A	B	C	D	E
1	Name	Type	Rows	Data_length	
2	AccessLog	MyISAM	1162	227004	
3	AdContracts	MyISAM	7	344	
4	AdPlacements	MyISAM	11	1404	
5	AdViews	MyISAM	27955	1321996	
6	ArtGrpLink	MyISAM	1277	31925	
7	ArticleGroups	MyISAM	204	12640	
8	ArticleTypes	MyISAM	0	0	
9	Articles	MyISAM	1293	473256	
10	AuthorArticle	MyISAM	346	11764	
11	Boilerplate	MyISAM	20	41652	
12	CommentGroups	MyISAM	1	44	
13	Comments	MyISAM	92	27396	
14	Control	MyISAM	1	8	
15	Detail nn	MyISAM	7290	370572	

**Figure 2. Viewing/editing a spreadsheet inside the OLE control**

You can even drag the docked toolbar off the window, as shown in Figure 3 (and of course you can redock the toolbar as well). Andy expects to add support for the full OLE container specification, including menu merging and other features. This is all pure Windows API code, so in standalone mode the EXE is only 19k. There will also be an option for API window support which will let you get rid of most of the remaining RTL windowing code when you compile in local mode.

	A	B	C	D	E
1	Name	Type	Rows	Data_length	
2	AccessLog	MyISAM	1162	227004	
3	AdContracts	MyISAM	7	344	
4	AdPlacements				
5	AdViews				
6	ArtGrpLink				
7	ArticleGroups				
8	ArticleTypes				
9	Articles	MyISAM	1293	473256	
10	AuthorArticle	MyISAM	346	11764	
11	Boilerplate	MyISAM	20	41652	
12	CommentGroups	MyISAM	1	44	
13	Comments	MyISAM	92	27396	
14	Control	MyISAM	1	8	
15	DetailLog	MyISAM	7290	370572	
16	Discussions	MyISAM	206	107436	

**Figure 3. The Excel spreadsheet with undocked toolbar**

You can also drop PDFs (see Figure 4) and PowerPoint presentations into the container.

The screenshot shows the OleWindow container displaying a PDF document. The toolbar at the top includes icons for file operations (save, print, copy, paste), editing (undo, redo, find), and navigation (back, forward, search). The left sidebar contains a 'Bookmarks' pane with a tree view of the document's contents, including 'clarionmag.com', 'Clarion Magazine', 'Interview: ClarioNET's Micha...', 'Wininet.DLL: Transferring Fil...', 'Do Not Adjust Your Browser', 'Report Redirection', 'How To Make Unplanned Se...', 'News', 'Remembering September 11', 'Designing Crosstab Reports', 'When a Datum is a Functio...', 'Self-Upgrading Applications', 'Finding Unused Variables', 'Designing Crosstab Reports', 'Normal Files, Normal Displa...', and 'The Clarion Magazine Free S...'. The main area displays the PDF content, which includes a header for 'Clarion MAGAZINE' and several articles. The status bar at the bottom indicates '1 of 83' pages.

**Figure 4. The OLE container showing a PDF**

[Click here](#) to see some of the OLE class methods. This isn't code you'll need to understand in order to use the classes, but it does provide a little insight into the kind of work Andy has put into this product.

The OLE classes will be a commercial product, but pricing and availability are not yet known.

---

*[David Harms](#) is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of Developing Clarion for Windows Applications, published by SAMS (1995). His most recent book is [JSP, Servlets, and MySQL](#), published by HungryMinds Inc. (2001).*

## Reader Comments

[Add a comment](#)

**Nice... Very Nice**

**Woo Hoo!**

**The attached code has some silly bugs in it from a copy /...**

**hi andy, please hurry up ! i will love AND buy it...**

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.



# Clarion MAGAZINE

## MySQL: InnoDB Tables And Transactions (Part 1)

by **Dave Harms**

Published 2001-10-11

In the spring of 2000 I wrote a [series of articles](#) on using the MySQL database server with Clarion. I intended to update that series this spring, but you know what happens to deadlines. In any case, there have been enough significant developments in the MySQL world that a new series is definitely warranted. This week I'll kick off with a discussion of newly-added transaction support and other related features; over the next month or two I'll also discuss converting TPS applications to MySQL.

Just to refresh your memory, MySQL is a fast, reliable, multi-platform, and free (under the GNU General Public License) SQL server. Developed specifically for speed and reliability, MySQL is missing some of the common SQL server features such as sub-selects, stored procedures, and triggers/server-side referential integrity. That may seem like a pretty long laundry list. But as Jim Gambon pointed out in the [follow-up](#) to that first series of articles, MySQL makes a more than adequate replacement for TPS files, as you can easily get Clarion to handle the critical RI issues.

**NOTE:** Sub-selects are slated for MySQL release 4.1 (4.0 is about to go alpha, and 4.1 is expected in December).

### Atomic operations

The lack of transaction support was a more serious problem. MySQL in its native state uses atomic operations instead of transactions. An atomic operation is any single table update. It's a bit like a mini-transaction: either the table update happens or it doesn't.

According to the [MySQL documentation](#), atomic operations can be three to five times faster than transactions. Atomic operations also guarantee that you won't get any dirty reads (a read of data in the process of being changed). The downsides are a) updates are on a per-table basis, so you can't group a bunch of updates together under one atomic operation, b) there's no way to explicitly roll

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information](#)

[Log In](#)

[Membership/  
Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



back an atomic operation (although you could manually undo whatever you just did), and c) because atomic operations prevent dirty reads, a table update will block a SELECT on the same table, which can have performance implications.

If this is how MySQL handles updates, then how do you get it to do transactions? Easy – just don't use MySQL tables.

Let me explain that. MySQL is a SQL DBMS, or database management system. Your application will (most likely) talk to MySQL using the MyODBC driver. However your app communicates with the server, MySQL receives those SQL statements, queries or updates data accordingly, and returns data if necessary. By default, MySQL calls its own low-level data access library to get at the data. But that isn't your only option.

## Table types

Just as one Clarion application can use multiple file/table types via different file drivers, so a MySQL server can work with multiple table types by plugging in the appropriate back-end library. As of MySQL 3.23, there are seven table types (that I know of) available: MyISAM, MERGE, ISAM, HEAP, BDB, InnoDB, and Gemini. Each of these table types has its own advantages and disadvantages, and several originate with companies other than MySQL AB.

### MyISAM tables

If you issue a CREATE TABLE statement in MySQL, without a TYPE attribute, MySQL will create a MyISAM table. MyISAM is an update to the original MySQL ISAM table type. MyISAM is, if you like, MySQL's native table type. Big (63 bit) files are supported on 64 bit operating systems, and all data is stored low byte first, which makes the data machine-independent.

### ISAM tables

ISAM tables are the original MySQL "native" B-tree ISAM format tables. Data is stored in machine-dependent format. This table type is deprecated (replaced by MyISAM) and is expected to disappear entirely before much longer.

### MERGE tables

New in MySQL 3.23.25, MERGE tables (also known as MRG\_MyISAM tables) are collections of *identical* MyISAM tables which you can use as a single table. Keys and columns must match among the merged tables. Why would you want to do this? Perhaps for a log file, where you can split the data by month and get better performance when looking at just one table, yet still search all tables as one if you need to. You can also selectively create merged tables without altering the source table definitions. If you have very large tables, you can locate different tables on

different disks, and treat them as a single table. As well, a merged table over a single table is effectively an alias. I haven't tested MERGE yet, but I'm considering implementing the ClarionMag server log as a set of merged tables.

### **HEAP tables**

HEAP tables are very fast tables created in memory. That means if you lose power, you lose the data, but you get terrific speed. HEAP tables are commonly used for temporary tables.

### **Gemini tables**

The Gemini table types are available in [NuSphere's release](#) of MySQL. Its worth noting that there have been [legal disputes](#) between NuSphere and MySQL AB. I don't have a lot to say about Gemini because I don't own a copy – NuSphere's release of MySQL with Gemini is a commercial product.

### **BDB tables**

The BDB, or BerkeleyDB table type is another of the MySQL table types which do not originate with MySQL AB. The Berkeley database system was created by Dr. Margo Seltzer and Keith Bostic in the early 90s; Selzter and Bostic later founded [SleepyCat](#). Software.

A patched version of BDB ships with the MySQL source distribution; you can't use the non-patched BDB with MySQL. Among other things, BDB tables provide transaction support, and are slightly larger and slower than MyISAM tables.

### **InnoDB tables**

As with the BDB table type, the InnoDB table type does not originate with MySQL AB. [InnoDB](#) is the brainchild of Heikki Tuuri, a Finnish developer who holds a PhD in mathematical logic from the University. Tuuri began development of InnoDB in 1994, and created the MySQL interface in collaboration with MySQL AB, between September 2000 and March 2001.

A relatively new table type for MySQL, InnoDB has been generating a fair bit of excitement among MySQL users. It offers a number of important enhancements, such as transactions, row level locking (as opposed to MyISAM table locking/atomic operations), and high performance on large volumes of data. InnoDB tables are kept in a tablespace, rather than in individual files for each table. InnoDB tables do take up quite a bit more space than MyISAM tables.

### **Choosing a table type**

Although I've been relatively happy with MySQL's speed (I use MySQL primarily as the database underpinning Clarion Magazine),

I have noticed occasional performance problems which I suspected were caused by updates blocking SELECT statements. This is a predictable situation given MySQL's use of atomic operations to maintain data integrity. InnoDB tables, on the other hand, lock at the row level, not the table level, and provide Oracle-style consistent reads. The InnoDB manual puts it this way:

A consistent read means that InnoDB uses its multiversioning to present to a query a snapshot of the database at a point in time. The query will see the changes made by exactly those transactions that committed before that point of time, and no changes made by later or uncommitted transactions. The exception to this rule is that the query will see the changes made by the transaction itself which issues the query.

(The manual also points out that consistent reads aren't always desirable; if your query needs to be absolutely sure that the records it sees have not been removed or altered since the query began, you can do a locking read by appending the phrase `LOCK IN SHARE MODE` to the `SELECT` statement.)

Consistent reads and row level locks certainly looked like useful improvements to my server installation, so I decided to go ahead and give InnoDB tables a try.

## Installing an InnoDB-capable MySQL

My first step was to upgrade my existing MySQL server to a binary version (as opposed to getting, and building, the source) that supports InnoDB tables. This version is called MySQL-Max, and like "regular" MySQL, is freely available under the GPL. Just go to [www.mysql.com](http://www.mysql.com) and look for the Versions box on the home page. You'll see a link to the current releases of key MySQL products, including MySQL-Max. The version I'm using is 3.23.42.

Upgrading my MySQL installation proved quite simple. First, I unpacked the MySQL-Max distribution under the `/usr/local` directory. In my case, the command was:

```
tar -zxvf /tmp/mysql-max*
```

I used the wildcard in the gzipped tar filename to save myself some typing. The full command would be

```
tar -zxvf /tmp/mysql-max-3.23.42-pc-  
-linux-gnu-i686.tar.gz
```

assuming the gzipped tar is in the `/tmp` directory and you're already in the `/usr/local` directory. This command unpacked all of the `mysql-max` binary distribution files into the `/usr/local/mysql-max-3.23.42-pc-linux-gnu-i686` directory.

The next step was to shut down the currently running MySQL server and switch everything to the new version. I executed this command:

```
mysqladmin shutdown -u root -p
```

Because I'd previously set a root password for mysql (always a good idea), I had to supply the root user id and the `-p` parameter. I could have supplied the actual password after the `-` parameter, but it's better to let `mysqladmin` (and any other utilities) prompt you, as this is more secure.

With the MySQL server shut down I was now in a position to switch servers. When I first installed MySQL on this particular server, the files all went in the `/usr/local/mysql-3.23.35-pc-linux-gnu-1686` directory. In keeping with the install instructions, I created a symbolic link to that directory, called `mysql`. That meant I could refer to the `/usr/local/mysql-3.23.35-pc-linux-gnu-1686` directory as simply `/usr/local/mysql`. So the first step to the migration was to delete the old link:

```
cd /usr/local
rm mysql
```

and add a new one:

```
ln -s /usr/local/mysql-max-3.23.42
    -pc-linux-gnu-i686 mysql
```

If this was a new installation, I'd next run the `mysql_install_db` script to create the default `mysql` and `test` databases, but because I already have a database set up this isn't necessary, or desirable. Instead, I simply copied all the database files to their new location:

```
cp /usr/local/mysql-3.23.35-pc-linux-gnu-1686/data/*
    /usr/local/mysql/data -Rvf
```

The `-Rvf` options, respectively, tell `cp` to copy recursively, display file names, and force the copy where necessary. Actually I could also have moved the database to the new location, but I wanted to keep a backup copy in case something went wrong.

I started up `mysqld` with this command:

```
/usr/local/bin/safe_mysqld -- user=mysql &
```

And `mysql` loaded up, and everything worked! Wonderful. Just to make sure everything was copacetic, I rebooted the server. Unfortunately, the MySQL server did *not* start up on the reboot. After a little investigation I discovered that I'd started up the server with code in my `/etc/rc.d/rc.local` file, as follows:

```
/usr/local/mysql/support-files/mysql.server start
```

A quick check of that particular file confirmed what I suspected: it wasn't marked as executable, as it was part of the new install. So I ran this command:

```
chmod +x mysql.server
```

from the `/usr/local/mysql/support-files` directory, and tried again. This time when the server rebooted, the MySQL server ran automatically. To confirm that I had the right version of MySQL running, I started the command line interface (CLI), and executed this command:

```
mysql> select version();
+-----+
| version() |
+-----+
| 3.23.42-max |
+-----+

1 row in set (0.00 sec)
```

The next step was to create an InnoDB database. Like Oracle tables, InnoDB tables live inside a tablespace, which is one or more physical files. To create the tablespace you add some appropriate settings to the `my.cnf` configuration file, and then you start the MySQL server. I'm running this particular server on a RedHat 7.0 Linux box, and I used the following settings in my `/etc/my.cnf` file:

```
[mysqld]
innodb_data_file_path = ibdata/ibdata1:1G
innodb_data_home_dir = /
set-variable = innodb_mirrored_log_groups=1
innodb_log_group_home_dir = /iblogs
set-variable = innodb_log_files_in_group=3
set-variable = innodb_log_file_size=50M
set-variable = innodb_log_buffer_size=8M
innodb_flush_log_at_trx_commit=1
innodb_log_arch_dir = /iblogs
innodb_log_archive=0
set-variable = innodb_buffer_pool_size=400M
set-variable = innodb_additional_mem_pool_size=20M
set-variable = innodb_file_io_threads=4
set-variable = innodb_lock_wait_timeout=50
```

I simply copied these settings from the MySQL online documentation, with a few exceptions. My `innodb_data_file_path` setting only specifies a single file for the tablespace, one gigabyte in size, while the example showed two tablespaces. I also changed the `innodb_log_group_home_dir` and `innodb_log_arch_dir` settings to suit my system, but I made the mistake of making these different directories. When I fired up MySQL, I saw the following in my error log (in `/usr/local/mysql/data`):

```

011004 12:54:43  mysqld started
InnoDB: Error: you must set the log group ←
  home dir in my.cnf the InnoDB: same as log arch dir.
011004 12:54:44  Can't init databases
011004 12:54:44  mysqld ended

```

**As it turns out, in this release of MySQL you have to specify the same directory for the log and log archive files. I corrected this problem, deleted the /ibdata/ibdata1 file (per the instructions regarding failed InnoDB initialization) and tried again. No joy – shortly after beginning to write the data file, InnoDB bailed on an operating system error.**

```

011004 13:11:43  mysqld started
InnoDB: The first specified data file ←
  /ibdata/ibdata1 did not exist:
InnoDB: a new database to be created!
InnoDB: Setting file /ibdata/ibdata1 ←
  size to 1073741824
InnoDB: Database physically writes the ←
  file full: wait...
InnoDB: operating system error number ←
  22 in a file operation.
InnoDB: Cannot continue operation.
011004 13:11:54  mysqld ended

```

**I couldn't find any documentation explaining the nature of the error, so I deleted the one data file again, and this time rebooted. When MySQL came up on the reboot, it printed the following information to the log:**

```

011004 13:17:53  mysqld started
InnoDB: The first specified data file ←
  /ibdata/ibdata1 did not exist:
InnoDB: a new database to be created!
InnoDB: Setting file /ibdata/ibdata1 ←
  size to 1073741824
InnoDB: Database physically writes the ←
  file full: wait...
InnoDB: Log file /iblogs/ib_logfile0 did ←
  not exist: new to be created
InnoDB: Setting log file /iblogs/ib_logfile0 ←
  size to 52428800
InnoDB: Log file /iblogs/ib_logfile1 did not ←
  exist: new to be created
InnoDB: Setting log file /iblogs/ib_logfile1 ←
  size to 52428800
InnoDB: Log file /iblogs/ib_logfile2 did not ←
  exist: new to be created
InnoDB: Setting log file /iblogs/ib_logfile2 ←
  size to 52428800
InnoDB: Doublewrite buffer not found: ←
  creating new
InnoDB: Doublewrite buffer created

```

```
011004 13:19:58 InnoDB: Started
/usr/local/mysql/bin/mysqld: ready for connections
```

Success! I had an InnoDB database running under MySQL, and I could begin creating InnoDB tables. In the next installment in this series I'll do some comparative testing between MyISAM and InnoDB tables, and take a look at InnoDB transactions.

---

*[David Harms](#) is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995). His most recent book is [JSP, Servlets, and MySQL](#), published by HungryMinds Inc. (2001).*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 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.



*This is an excerpt from the IRC chat on Saturday, October 6, 2001. Andy Ireland, of **PlugWare.Com Solutions, Ltd** talked about interfaces, COM objects and how to use them in Clarion. Side conversations, parts and joins and other notices are removed for clarity. Also, the conversation is reformatted to read like an interview.*

Andy: I'll be able to start the ActiveX SDK next weekend depending on how quickly I get this other job done. I've got most of the interfaces defined. I will start with the basic support. COM objects I think I can do now.

*Q: ActiveX SDK? That's something to look forward to.*

A: Yes, after I ship the container support. I have a test app with the web browser control running in it in Clarion, all hand coded in Clarion classes. Pure API.

ActiveX, COM and OLE are the things I want to concentrate on getting good support for into Clarion. COM is done, ActiveX use is almost done, creation of COM and ActiveX objects next and then full-blown OLE support.

*Q: Andy - OK, let's cut to the chase. What can the average Clarion developer do with these classes?*

A: Use ActiveX controls without them going bang! Uses COM with relative ease, then eventually create ActiveX and COM objects.

*Q: In what context would it be good to have an ActiveX control in my application?*

A: There are many more 3rd party products then at your disposal.

*Q: Like what kind of "zing" does this bring to my apps that my clients would like?*

A: Anything from calendar controls to imaging and mapping controls to full-blown report engines etc. With a MicroSoft feel to the UI unlike many [existing] market products.

*Q: Sheridan Calendar controls.*

A: Exactly.

*Q: Imaging controls, graphing controls, I even know of a neural net solution that's ActiveX.*

A: I could do a wrapper for GCAL Pro really quickly for instance using the ActiveX.

*Q: OK, this sounds good. I am tired of looking at battleship gray.*

A: It took me 5 minutes to do the web browser control. Here's the example code (not many lines):

```

OleWindow WINDOW('OleWindow'), AT(, , 205, 179), SYSTEM, GRAY, RESIZE, IMM
    END
OleIniter          COLEIniter
WebBrwsr          WebBrowser
CtrlSite          &COLECtrlSite
CtrlContainer     COLECtrlContainer
fBrowserCreated   bool
ResizeCount       long

code
if OleIniter.IsInitialised()
    open(OleWindow)
    accept
        case event()
            of event:openwindow
                CreateBrowser
            of event:sized
                if fBrowserCreated and ResizeCount > 0
                    CtrlContainer.Resize().
                    ResizeCount += 1
                end
            end
        end
    end
end
if ~CtrlSite &= null
    dispose(CtrlSite).

```

and then...

```

CreateBrowser procedure
rcWindow      like(_RECT_)
szURL         cstring('www.microsoft.com')
ProgID        CStr
szProgID      &cstring
lpszProgID    long
hr            HRESULT, auto

code
hr = ProgIDFromCLSID(address(__IWebBrowser), lpszProgID)
if hr = S_OK
    if ProgID.Init(lpszProgID, true, 0, false)
        szProgID &= ProgID.GetCStr()
    end
    CoTaskMemFree(lpszProgID)
end
GetClientRect(OleWindow{prop:clienthandle}, rcWindow)
if CtrlContainer.Init(OleWindow{prop:handle}, OleWindow{prop:clienthandle})
    CtrlSite &= new COLECtrlSite
    CtrlSite.Init(CtrlContainer, OleWindow{prop:clienthandle}, rcWindow)
    hr = WebBrwsr.Init(CtrlSite)
    if hr = S_OK
        hr = WebBrwsr.Navigate(szURL)
        fBrowserCreated = true
    end
end
end

```

See? Not difficult at all. Ignore the ProgID code, was testing the looking up of names from IID's so I could instantiate based upon simple name such as webbrowser.2 or whatever its name is.

*Q: OK, really dumb question: How does this fit with an ABC template generated application? How much work to add these features?*

A: No templates planned just yet. The classes are ABC compliant.

*Q: Will I be able to receive events from an IE control?*

A: Yes. I do already.

*Q: Andy - OK, then assuming this code is complete, functional and perfect, is it a lot of work for the average Clarion developer to insert into their app files?*

A: No, it is simple, but can be [complex] if they want to take complete control and override the default code i.e. pass in their own control container interfaces for custom handling.

*Q: Andy, I would love to see it. Especially if I can get events from IE.*

A: Optionally pass in the events Interface or derive the OnEvent method. Events from IE are a trivial thing. You guys want to know how it works? I'll be happy to talk about it.

*Q: Andy; I'm game. Lets see some code.*

A: Ok, the key is something called an Event Sink. What happens is your application registers it's event sink (an interface in your code that the ActiveX can talk to). Actually there can be many but usually only one is called the primary. When the control is created it talks to an interface called IoleClientSite.

It asks this interface for the event sink interface. Now, here's the crux. If you do not support the native interface, in the case of the webbrowser control it's IwebBrowserEvents.

You give it an IDispatch interface. This interface is a completely generic interface. This is what Clarion always uses. It knows nothing about your ActiveX and therefore has to ask the control questions such as, "This eventID is what?"

Somewhere in this process, the RTL is screwing up. My code allows you to either use this IDispatch or pass in the native interface. If you pass in the native interface, it calls into it, passing you the info as you expect it. If not it is passed in a generic format and needs some processing.

Ok, I'll pause for questions. Shall I explain what the interfaces are and do?

*Q: Please!*

A: OK, there are two main interfaces in COM: IUnknown and IDispatch. IDispatch is derived from Iunknown. So IDispatch has all the methods of IUnknown and then also it's own.

*Q: Andy - question. The "I" is for "Interface, right?*

A: Yes

*Q: Please explain the Dispatch and Unknown labels.*

A: The IUnknown is the generic interface for talking to a COM object and it contains the following....

```
IUnknown
QueryInterface
AddRef
Release
Interface
procedure(REFIID riid, *long ppvObject), long, pascal
procedure(), long, pascal
procedure(), long, pascal, proc
end
```

*Q: IUnknown meaning "I don't know what it could possibly be, but give it to me anyway"?*

A: No, IUnknown meaning "I do not know or care what the object is but I need to be able to talk to it in some generic manner."

*Q: OK. That makes sense.*

A: How else could I say..."Give me something if you support it"?

*Q: And what does Dispatch mean?*

A: That where QueryInterface comes in, but hold one... not finished with IUnknown.

*Q: OK.*

A: [To continue,] AddRef and Release allow me to keep a count of how many instances of code are using my interface.

*Q: OK, that makes sense.*

A: [Further,] when the release method is called and if the count hits 0, the object is destroyed and cleaned up.

*Q: Kind of like what the old CheckOpen function did.*

A: The QueryInterface allows me to ask for a supported interface but with open files. I.e. if you support a given interface, give it to me. The ref count on that interface gets incremented. When I am done with it, I release it. [This is] easy so far.

*Q: Yes, perfect sense. Anyone not follow?*

A: All COM objects, be they ActiveX, COM or OLE support this. They must. Ok, so where does IDispatch come in?

IDispatch extends this model to allow me to give an object commands. For instance set/get a property, execute a method or handle an event. The interface is:

```
Idispatch          interface(IUnknown)
GetTypeInfoCount  procedure(*long pctinfo),HRESULT,pascal
GetTypeInfo       procedure(long iTInfo, long lcid, long ppTInfo),HRESULT,pascal
GetIDsOfNames     procedure(long riid, long prgszNames, long cNames, |
                  long lcid, long prgDispId),HRESULT,pascal
Invoke            procedure(long dispIdMember, long riid, long lcid, short wFlags, |
                  long pDispParams, long pVarResult, long pExcepInfo, |
                  long puArgErr),HRESULT,pascal
end
```

*Q: So IUnknown says "give me your supported interface and I will track the instances for you.*

A: yes and no. IUnknown allows me as a consumer to ask and receive an interface the object that gives me the interface increments the ref count. I release it when done. If my code is the object, I increment, the consumer releases

*Q: IDispatch says "I will handle any communication you need", correct?*

A: You are correct, but IDispatch goes a little further. You can also ask it to give you the detailed information about the control or object such as what it's interfaces are and the prototypes on the fly as this is stored in the registry.

This allows you to dynamically create interfaces as needed that match the object rather than being generic. Generic comes at a cost -- it is slower. This is because everything must then be interpreted instead of calls made directly into the relevant native interfaces.

*Q: So I (the developer) do not really need to know the interface makeup?*

A: Right. Is this clear so far?

*Q: OK, here is the next question. If I do not know the context or makeup of an interface, how do I use it? Or even if I want to use it?*

A: The difference is called late binding for generic, early binding for native. You need to know what some of the info is which is why IDispatch supports you asking. For example, lets say I use the WebBrowser control.

If I want to trap an event, I need to know what the ID for that event is whether I like it or not, for it to be useful. Otherwise I may as well not bother. So late binding is what VB uses so that I do not need to code anything special to use a control. But there are some things I either must know or must ask otherwise my object just looks pretty on the screen.

*Q: Like which events are even relevant for the control?*

A: Yes. Here is another example. If I use the web browser control, how do I go to a web page if I do not know how to ask the question? Therefore as an implementer I still must know implementation details specific to the control. The difference is, my container does not need to know this therefore only my apps business logic changes, not my classes.

*Q: Andy - would you give an example of one event for a web browser and how it would be used?*

A: Lets say I ask the browser to navigate to [www.softvelocity.com](http://www.softvelocity.com). When it does this, it fires two events, one before navigation, and one after. It actually fires more but this is for simplicity. The one before allows me to return true or false to allow navigation or not i.e. a link may have been clicked but I want to trap and disallow this is where I trap the event change the value and return.

It's easy. But we can be more creative. Lets say I create a web page. On this page is a set of links, which are controls. Each of those represents valid controls in a screen editor. When one of those is clicked, I trap the event, stop the navigation click on the window, it sends a msg and creates that control.

Now, I have created a window formatter whose control list is an html page and it uses dynamic modules to define the controls so not only can I add controls to the system but also amend the html accordingly.

So the events can be used in many ways. Is that clear? Likewise that could control navigation in an app and be the menu, which can be changed without changing code.

OK, so where are we at? We know what the base interface is for COM objects.

We know all objects be them ActiveX, OLE or COM are COM objects. It's all MS semantics. COM added some features but for arguments sake this is the case.

So where does a control differ? A control has a GUI and it needs to know where, how and on what it is displaying itself. This requires a more extensive conversation, i.e. more info, more advanced data, and if it relied on the two bases interfaces, it would be so chatty that it would run slower than water in the north pole.

So GUI controls (ActiveX controls) add some interfaces for this purpose. But the interfaces are needed on both sides of the conversation, so it talks to me and I respond in kind. If it talks German, so must I.

By that I mean if it has a GUI conversation, I need to support that. It asks where it is displaying itself and how to present itself, I respond with a window, a frame and some details of the format i.e. display as a BMP on a surface which is for a printer or for a screen.

Additionally, I tell it how big it can be and where it can and can't draw. For this I need extra interfaces. On the container side these include:

```
IoleInPlaceFrame
IoleInPlaceUIWindow
IoleInPlaceSite
IoleClientSite
IAdviseSink
IoleControlSite
IDispatch (Ambient properties)
IPropertyNotifySink
IDispatch (Events)
```

These need to be implemented in an ActiveX container and amounts to about 2000 lines of code give or take. But it depends on the amount of support you wish to allow, not all of this is mandatory.

*Q: The problem, I think, is that there is so much of it. It is hard to take a piece of data, absorb that, take another etc.*

A: Then we should break up into lessons if people are interested. COM is easy, there is just lots of it. Here's a snippet of code:

```
COLECtrlSite.CreateCtrl procedure(long rclsid, long riid, |
                                long renderopt, *long pvObject)
hr          HRESULT, auto
lFormatEtc like(FORMATETC)

code
? assert(~self.InPlaceSite &= null)
? assert(~self.ClientSite &= null)
? assert(~self.AdviseSink &= null)
? assert(~self.AmbientDispatch &= null)
? assert(~self.EventDispatch &= null)
? assert(~self.Storage &= null)
if (~self.InPlaceSite &= null) and (~self.ClientSite &= null) and |
    (~self.AdviseSink &= null) and (~self.AmbientDispatch &= null) and |
    (~self.EventDispatch &= null) and (~self.Storage &= null)
lFormatEtc.cfFormat = CF_BITMAP
lFormatEtc.ptd = 0
lFormatEtc.dwAspect = DVASPECT_CONTENT
lFormatEtc.tymed = TYMED_GDI
lFormatEtc.lIndex = -1
hr = self.Storage.Init()
if hr = S_OK
    hr = OleCreate(rclsid, riid, renderopt, 0, self.ClientSite.IoleClientSite, |
                  self.Storage.GetStorage(), pvObject)
if hr = S_OK
    self.pObj = pvObject
    hr = self.Advise()
    if hr = S_OK
        hr = self.SetEventDispatch()
```

```

        if hr = S_OK
            self.bInitialised = true.
        end
    end
end
end
return hr

```

This creates a control, i.e. create an ActiveX before it displays however, lots of conversations take place so one must implement everything before creation will occur.

See where I tell it how to display? For a window, this is the simplest way to setup display info. I allow this to be overridden so COM experts can do more complex handling.

See the OleCreate line. This is where I give it my IOleClientSite interface. It asks this for whatever it needs from me. Think of this interface, as it's main conversation conduit for it to talk to me. I also give it a storage interface for it's caching and it returns its base interface in pvObject. I hold onto this interface and release at the end this is my way to talk to it.

Therefore, a conversation is setup. Then I must simply support whatever it needs. Advise is where I tell it what my *AdviseSink* is and *SetEventDispatch* is where I tell it what my event interface is.

When an event occurs, it passes it to me via this, and then it starts getting really heavy! But if you understand the basics, it is very simple just lots to do.

Are there any final questions? We'll end the lesson here. Shall I continue next week?

*Q: Please!*

A: I'll do a brief recap on this then go further.

*Q: But in essence, I created a communication bridge (interface) and then send messages back and forth.*

A: Yes

Maybe leave GUI controls and talk about COM, DCOM and something like ADO, but several of them with different roles in the grand scheme of things.

*Q: Once I know the messages, then I have my conditional code.*

A: Yes, for instance, Navigate.

*Q: Just trying to make as simplistic overview as I possibly can.*

A: This is just a method ID. It says a method of x address is mapped to this equate, when I ask for it, call it.

*Q: I like that.*

A: That is what late binding does hence why late binding is slow and why VB apps tend to be slow. Early binding is yes I know the interface, in fact I have asked for it explicitly and I will call directly, hence just a call in Asm.

*Q: If it is just a reference to an address, does that not mean an extra lookup? So why so slow if this is the case?*

A: Yes, but also passing the params in a generic way i.e. as a safearray of variant structures hence the extra work involved.

*Q: I see, the extra data conversion?*

A: This must be marshaled. I will leave marshalling for another time as is complex.

The crux though is this, marshalling is essentially a way to wrap the data up and pass it cross process boundaries, etc. Hence there is a certain amount of synchronization going on.

Oh, for those that don't know, a variant is a group that allows you to wrap up a data type in a generic way like an ANY variable.

*Q: Ah! That safearray problem was the big issue with my CW/CorelDraw COM interface. Never did find an understandable solution.*

A: I posted code on comp.lang.clarion to resolve that plus the equates etc and prototypes plus some asm code for calling methods that need a variant passed by value as CW does not support this.

*Q: Andy - Thanks a lot.*

A: You're all welcome. Always willing to help the community. Oh and a cheap plug --Plugware Solutions.com Ltd will be releasing an ActiveX control container library and all sources as a product. It is a complete replacement of the OLE support in the RTL. Initially it will support ActiveX, but later, full-blown OLE container support. Then onto ActiveX / COM object creation SDK for Clarion.

*Q: Andy - Not that it matters much, but what is the \$\$\$?*

A: I've, not decided. But I will do two versions, one without source (cheap), one with (expensive'ish). COM I will post for free.

*Q: Andy - I thought you did this for ADO. COM you reserved.*

A: I need to put the COM classes into OpenSource. Yes, COM is mine, but it ships as source with the ADO so I may as well make it OpenSource.

*Q: Andy - Don't forget the Clarion Magazine open source thingy.*

A: It's for the community. Hope this talk was of value to people. Oh, and as another shameless plug, my company does consulting work for anyone looking for advanced features.

Group: Andy - thanks so much for the data.



# Clarion MAGAZINE

## MySQL: InnoDB Tables And Transactions (Part 2)

by **Dave Harms**

Published 2001-10-17

In [Part 1 of this series](#) I discussed the various table types currently available to the [MySQL](#) database server, and explained how to install a version of MySQL that supports the [InnoDB](#) table types. This week I'll create an application to test transactions on MySQL, using InnoDB tables.

For some time I've been using the 2.50.33 release of the MyODBC driver. I decided it was time to get current, and so I went to the [MySQL home page](#) where I discovered that I was six revs behind the times! In fact, support for transactions was only added to MyODBC in release 2.50.37, so I didn't have a choice – I needed the updated driver to use InnoDB transactions. As I already had gone through the full MyODBC install in the past, I downloaded the "DLL only" install.

**NOTE:** If you're running Clarion 5.0 or earlier, you may not be able to use the current releases of the MyODBC driver (and therefore you cannot use transactions with InnoDB tables). The problem is that the Clarion ODBC driver uses a different sequence of ODBC statements than that expected by MyODBC. In Clarion 5.5x, you can tell the driver to use a MyODBC-compatible sequence by specifying the `/BINDCOLORDER=2` driver string. Ron Schofield (<http://www.openclarion.org/cgi-bin/downloads.cgi?dir=/mysql>) does have a Clarion 5.0-compatible driver available, but its currently based on the 2.50.31 release.

To install just the new MyODBC DLL I extracted the zip contents to a temp directory, and then ran `INSTALL.BAT`. Actually I thought it a bit odd that the batch file was able to copy the files over, as I'd been using the MyODBC driver just prior to the update and expected the DLL would be marked as still in use. I checked the ODBC administrator and it reported that the DLL was still the old version. So I did what I usually have to do in a situation like this: I rebooted and tried again. The batch file ran again with no errors and still the ODBC administrator reported the old version.

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



At that point I checked the batch file and found it was copying files into the Windows `system` directory; I'm running NT, and those files needed to go into the `system32` directory. A quick search and replace on the batch file, one more attempt, and MyODBC was finally updated.

If you haven't used MyODBC before then you will want to download the full install – this will set up MyODBC in the ODBC administrator; as far as I know, that install doesn't suffer from the same problem as the DLL update batch file.

## Creating tables

Once you have MyODBC installed, you're ready to create the tables. There are a number of ways you can go about this, including using Roberto Artigas' templates for converting from TPS to MySQL (Roberto regularly posts releases to the `SoftVelocity.topic.thirdparty` newsgroup).

I'll have a look at Roberto's code in a later article; this time around, I'll show how to create a few simple tables from scratch. These tables, called `Parents` and `Children`, will rather obviously contain parent and child records, in a one to many relationship. You'll also need a third table to support server-side autoincrementing (more on that later). Listing 1 shows the SQL code used to create a new database called `transact` and populate it with the three tables.

### Listing 1. Creating the transact database's tables

```
CREATE DATABASE transact;
USE transact;

CREATE TABLE Parents(
  ParentID INT NOT NULL AUTO_INCREMENT,
  LastModified TIMESTAMP,
  PRIMARY KEY (ParentID))
TYPE=INNODB;

CREATE TABLE Children(
  ChildID INT NOT NULL AUTO_INCREMENT,
  ParentID INT NOT NULL,
  SomeWeirdField INT DEFAULT 0,
  LastModified TIMESTAMP,
  PRIMARY KEY (ChildID),
  INDEX ChildrenIdx_ParentID (ParentID))
TYPE=INNODB;

CREATE TABLE lastinsert(ID INT) TYPE=INNODB;
```

After you create the tables, you need to tell MySQL to allow your application to access the database. Assuming the server and the application are running on the same machine (called `localhost`), and the user id and password are both `transtest`, the following

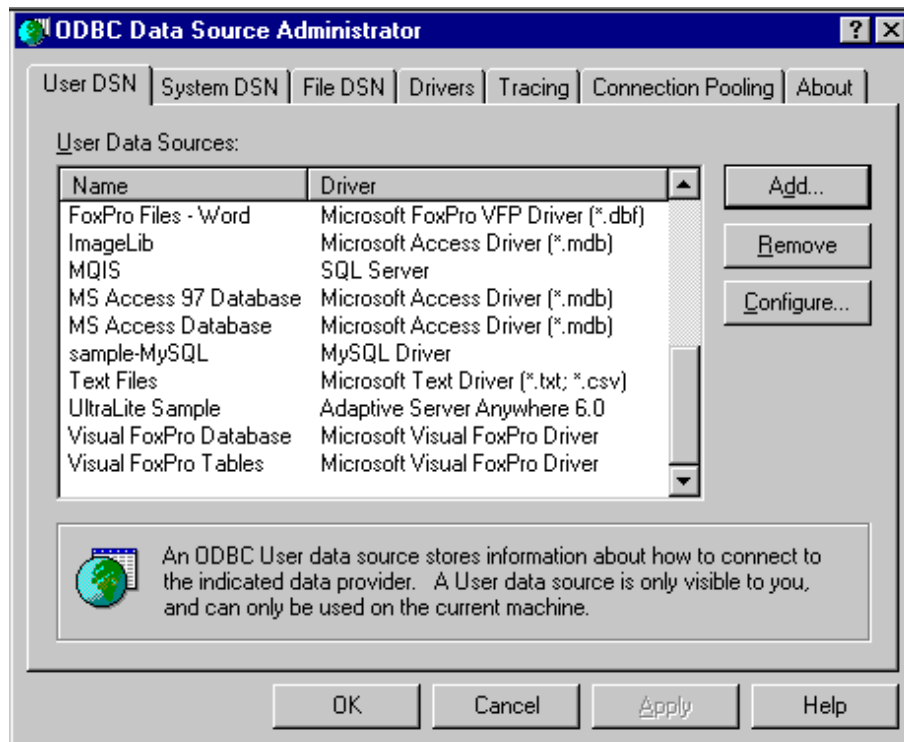
code will do the trick :

```
GRANT ALL ON transact.* TO transtest@localhost
  IDENTIFIED BY 'transtest';
FLUSH PRIVILEGES;
```

The GRANT statement will create entries in the user and db tables in the mysql database; MySQL uses these tables (and optionally several others) to control who do what to which tables and columns. By issuing a GRANT ALL you are allowing this user full control over the transact database; in reality you'd probably want to restrict rights a bit, particularly since GRANT ALL allows the user to read physical, non-MySQL files on the server!

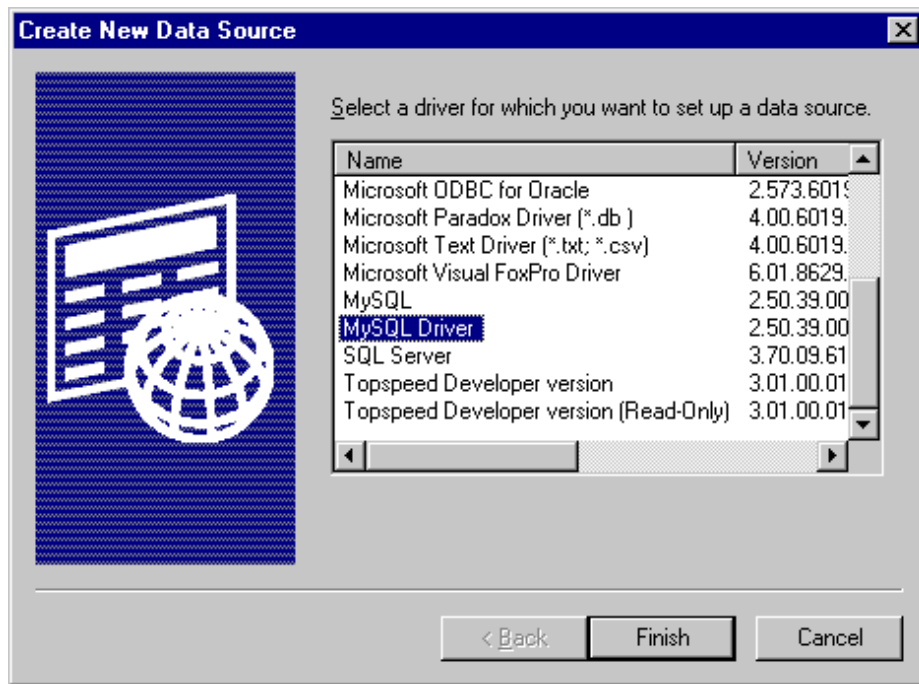
## Setting up a data source

Let's say you have MyODBC installed, and you now have some tables. The easiest way to build an application around those tables is to just import them into your application, and to do that you need to define a data source. Go to the Control Panel and open the ODBC Administrator. On the User DSN tab (shown in Figure 1) click on Add.



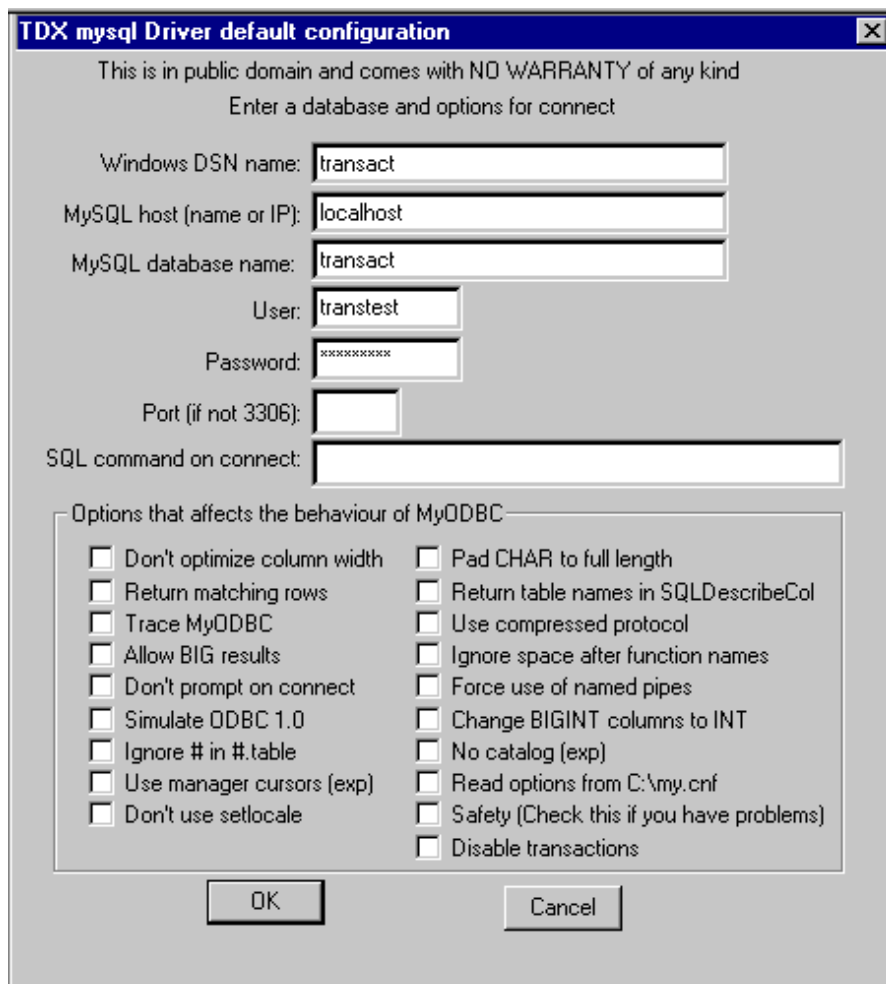
**Figure 1. The ODBC Administrator User DSN tab.**

From the New Data Source window (Figure 2) select the MySQL driver. As you can see, I have two entries. I don't know why - maybe it was that botched installation attempt.



**Figure 2. Selecting the MySQL driver**

Figure 3 shows the configuration settings for the MySQL driver. The Windows DSN name is the name your application will use to access the database. The host is the name or IP address of the computer that is running the MySQL server.



### Figure 3. MySQL configuration settings

## Importing the table definitions

Now you're ready to import the table definitions into your Clarion application. Create a new dictionary, and with that dictionary open select File|Import Table. Choose the ODBC driver, and when you're presented with a list of data source names, choose the one you created for your MySQL database (in this example, it's the DSN called `transact`). Click on Next and you should see the list of available tables. Go through the import procedure three times until you have all three of the `transact` tables in your dictionary.

I have noticed occasional problems with importing MySQL tables, and the `Children` table is a typical example. When you import this table you have to examine the list of field names to make sure that all have imported correctly. On my system, the `ParentID` field does not. I have to correct the field name (its an unprintable character after import), *and* I have to go to the Attributes tab and set the External name to `ParentID` as well. If the external name of each field doesn't match the name of the corresponding table column, you'll get an invalid record declaration error when you attempt to open the table with your Clarion app.

There's one other dictionary change to make. For at least one of the tables set the following in the Driver Options field:

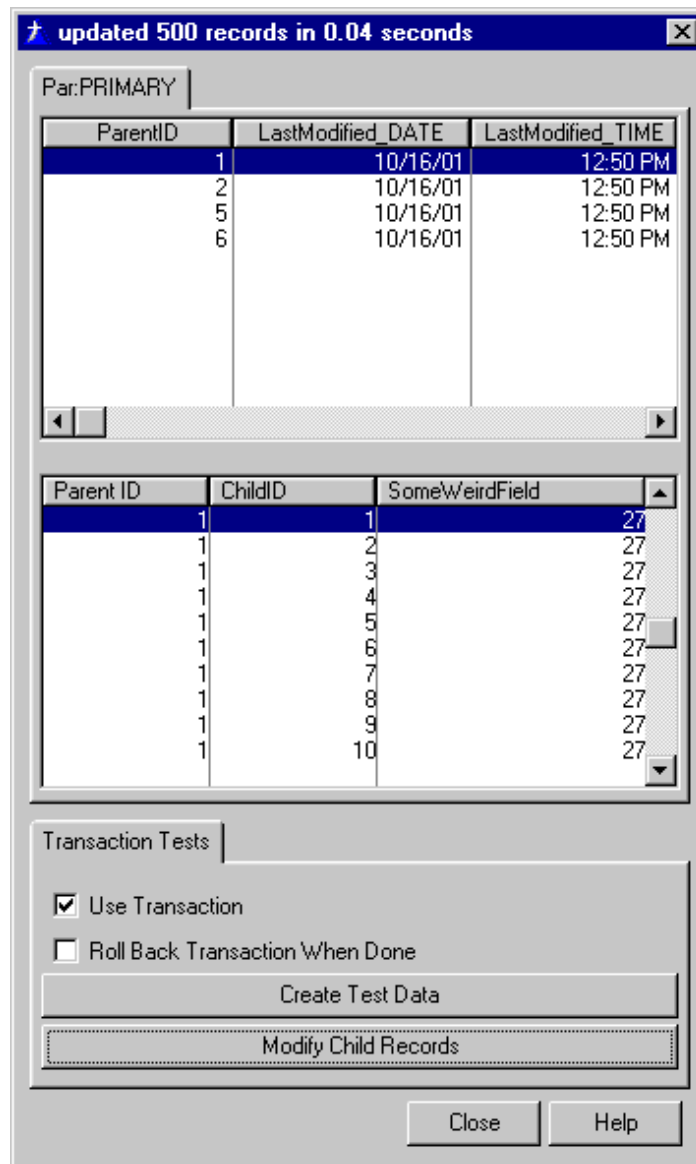
```
/BINDCOLORDER=2
```

This statement tells the Clarion ODBC driver to use a sequence of ODBC commands that is compatible with the MyODBC driver.

Now you can wizard up an application based on the dictionary and start playing with the data. (Or you can just download the example app at the end of this article.)

## Using transactions

Figure 1 shows the example application I used to test transactions with MySQL. Keep in mind that this application just demonstrates that transactions *work*. I'm not making any pretense of benchmarking here, although I have added some very rudimentary timings of basic operations.



**Figure 4. The transaction test application.**

The example application lets you create some parent and child data inside a transaction. It also lets you decide whether you want to place that data creation inside a transaction, and if so, whether you want the transaction to complete successfully, or to be rolled back. Here's the data creation code:

```

starttime = clock()
if UseTransaction then logout(1,Parents,Children).
setcursor(cursor:wait)
setnull(Par:ParentID)
if access:Parents.insert() = level:benign
  lastinsert{prop:sql} |
    = 'delete from lastinsert'
  lastinsert{prop:sql} |
    = 'insert into lastinsert (id) ' |
    & 'values (last_insert_id()''
  lastinsert{prop:sql} = 'select * from lastinsert'
access:lastinsert.next()
ParentID = las:ID

```

```

    y = random(1,100)
    loop x = 1 to maxRecs
        !access:Children.PrimeAutoInc()
        Chi:ParentID = ParentID
        Chi:SomeWeirdField = y
        setnull(Chi:ChildID)
        !0{prop:text} |
            = 'creating entry ' & x & ' of ' & maxRecs
        access:Children.Insert()
    end
end
if useTransaction
    if AbortTransaction
        rollback()
    else
        commit()
    end
end
endtime = clock()
setcursor()
0{prop:text} = 'processed ' & maxRecs |
    & ' records in ' & (endtime - starttime)/100 |
    & ' seconds'
ThisWindow.Reset(true)

```

**And here's the data modification code:**

```

starttime = clock()
if UseTransaction then logout(1,Children).
setcursor(cursor:wait)
y = random(1,100)
Children{prop:sql} |
    = 'update Children set SomeWeirdField=' & y |
    & ' where ParentID = ' & Par:ParentID
if useTransaction
    if AbortTransaction
        rollback()
    else
        commit()
    end
end
endtime = clock()
setcursor()
0{prop:text} = 'updated ' & maxKids |
    & ' records in ' & (endtime - starttime)/100 |
    & ' seconds'
BRW5.ResetFromFile()

```

**There are several points of interest in both blocks of code. First, although I'm calling LOGOUT on two tables, only one is necessary, in my experience; this triggers the call to the ODBC SQLTransact function.**

**Second, the data creation code relies on server-side autoincrementing of the primary key values, which speeds up**

inserts considerably as the Clarion application doesn't have to issue a SELECT before each insert to get the highest used key value. Server-side autoincrementing introduces a new problem, however. If you insert a parent record, you don't immediately know what its primary key is, because that's assigned by the server. You need that value for the foreign key in the child record, but it isn't yet in the parent record buffer. In MySQL, the `last_insert_id()` function will return the most recently-created autonumber primary key for the current connection.

It would be nice if you could return the value of a MySQL function directly using `PROP:SQL`, but instead you have to store that value in another table, and then retrieve it like any other data. That's what the `LastInsert` table is for. The following code uses `PROP:SQL` to remove any existing record from that table, adds a new record with the just-assigned autoincrement id, and returns that value with a `NEXT()` on the `LastInsert` table. Of course, it's still possible this code could fail in a multi-user situation, since others could be trying to change the data in the `LastInsert` table at the same time. You could add a `UserID` field to the `LastInsert` table, and keep one record per user instead of just one record.

```
if access:Parents.insert() = level:benign
  lastinsert{prop:sql} |
    = 'delete from lastinsert'
  lastinsert{prop:sql} |
    = 'insert into lastinsert (id) ' |
      & 'values (last_insert_id())'
  lastinsert{prop:sql} |
    = 'select * from lastinsert'
  access:lastinsert.next()
  ParentID = las:ID
```

So much for the code; how does the application run?

## Testing transactions

In my test installation, with MySQL Max v. 3.23.42 and MyODBC 2.50.39, transactions work just fine. The database server is a Celeron 400 running RedHat 7.1 with two IBM 20 GB drives mirrored on a 3Ware Escalade controller; my test application is running on an NT box on a 100BaseT network.

A test insert of 500 records, with transactions enabled, takes almost exactly one second (including creating the parent record and getting the autoincrement ID). The time is the same whether you commit or rollback the transaction – this is, I assume, because the server caches the writes. If, however, you turn off transactions, InnoDB inserts take quite a bit longer, as the server flushes each write. Instead of one second, the operation takes almost eight seconds!

Mass updates (changing one field in all 500 child records to a new value) is speedy, as you'd expect. On my installation this operation



takes about .04 seconds.

You can easily compare InnoDB tables to MyISAM tables: just run the following script against your transact database:

```
DROP TABLE Parents;
DROP TABLE Children;

CREATE TABLE Parents(
ParentID INT NOT NULL AUTO_INCREMENT,
LastModified TIMESTAMP,
PRIMARY KEY (ParentID))
TYPE=MYISAM;

CREATE TABLE Children(
ChildID INT NOT NULL AUTO_INCREMENT,
ParentID INT NOT NULL,
SomeWeirdField INT DEFAULT 0,
LastModified TIMESTAMP,
PRIMARY KEY (ChildID),
INDEX ChildrenIdx_ParentID (ParentID))
TYPE=MYISAM;
```

These definitions are identical to the original definitions except the table type is MYISAM instead of INNODB. Now it won't make any difference whether you enable or disable transactions, since MYISAM tables don't support transactions. MYISAM mass insert speed in the test application is comparable to INNODB with transactions enabled. Mass updates are several times faster than INNODB mass updates (on the order of .01 second to update all 500 records with one new field value instead of .04 seconds).

Michael "Monty" Widenius recently indicated on the MySQL mailing list that there is a benchmark page forthcoming (on the MySQL site) which will compare the MyISAM, InnoDB, and HEAP table types, and will interpret the results. He also indicated that if you have a lot of possible conflicts between writes/updates/selects, i.e. where any one statement could take a lot of time, InnoDB should be faster than MyISAM.

If you're doing just a lot of data retrieval, however, MyISAM should be faster. Each MyISAM table is stored in two physical files, one for the table's indexes, and another for the table's data. To retrieve a record, the library code looks at an index and then gets a record by data offset. In an InnoDB database the primary key and row are stored together, and additional keys are stored as that key plus the primary key. The result is that a fetch on just the primary key is probably faster with InnoDB, but slower for fetches on secondary keys since they require an additional lookup on the primary key. "Big rows" are also expected to give poorer performance than on MyISAM tables.

## **Which do I choose?**

It seems clear that for business software development, InnoDB

tables, with row-level locking and transactions, are a better choice than MyISAM tables. Although I can't yet done any real-world comparisons, I would also expect comparable or better performance out of InnoDB tables with the kinds of applications most Clarion developers create.

InnoDB tables are also growing foreign key support, which should be ready for prime time shortly. As of InnoDB version 3.23.43b you define foreign keys in the CREATE TABLE statement as follows:

```
CREATE TABLE parent(
  id INT NOT NULL,
  PRIMARY KEY (id))
TYPE=INNODB;

CREATE TABLE child(
  id INT,
  parent_id INT,
  INDEX par_ind (parent_id),
  FOREIGN KEY (parent_id) REFERENCES parent(id))
TYPE=INNODB;
```

There are no delete cascades at this time, so I assume that these definitions represent a restrict constraint, but I haven't tested any of this yet. See the InnoDB [manual](#) for more information.

The downside to InnoDB is its relative youth within the MySQL environment. Although Heikki Tuuri began developing InnoDB in 1994, the MySQL integration only began about a year ago. Despite this InnoDB has an excellent reputation, and is used live by a number of prominent sites, most noticeably [Slashdot](#). You also don't need to convert your whole database - you can easily mix InnoDB tables with MyISAM tables in the same database.

I'll be voting with my own code: I expect to migrate ClarionMag from MyISAM tables to InnoDB tables over the next few months. If you decide you want to try InnoDB tables, I strongly encourage you to check the MySQL and InnoDB sites regularly as bugs are occasionally reported, and from what I've seen quickly fixed.

[Download the source](#)

---

*David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with with Ross Santos of Developing Clarion for Windows Applications, published by SAMS (1995). His most recent book is [JSP, Servlets, and MySQL](#), published by HungryMinds Inc. (2001).*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

**INVEST**  
in your own abilities

**Clarion**  
magazine

published by  
**CoveComm Inc.**

# Clarion MAGAZINE

## Review: CapeSoft File Explorer v2.1

by **Tom Hebenstreit**

Published 2001-10-19

Information comes packaged in many forms, and for raw database access Clarion is an awesome tool. Quite often, though, I find myself wanting to incorporate other "pre-packaged" bits of information into my programs as well. By that I mean information in other common file formats such as HTML (web pages), PDF (Adobe Acrobat), WAV (audio), AVI (video) or even advanced formats like Macromedia Flash animations. All offer features well beyond the usual static display that the average Clarion program exhibits, and give you an enormous amount of flexibility in presenting information in a more forceful manner.

Tutorials, demos, sales tools, wizards, program help - all can benefit from an infusion of dynamic or pre-formatted content. Also, by using these common file formats you have a lot of freedom in choosing the tools you use to create the files to display. Even better, most of the programs associated with these file formats also provide OCX/ActiveX/OLE viewer controls that other programs can access. Yes, even Clarion.

The primary obstacle to using these OCX controls in the past, though, has been the poor level of Clarion OCX/ActiveX/OLE support. That and the daunting amount of research and tinkering it takes to figure how to actually implement a given ActiveX control in a Clarion program (unless you are like Jim Kane and like to program in Assembler just for grins).

With the arrival of the much more stable Clarion 5.5, though, long-time Clarion third-party vendor CapeSoft now had the platform they needed to solve the second half of the puzzle - making the use of those controls template-based and hence, easy.

### Major Features

File Explorer lets you incorporate the following types of files and features into your programs:

- HTML (View only). The common language of the World Wide Web, HTML files can be loaded either from the local hard drive or from the Internet (or both, via hyperlinks). Templates are

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information  
Log In](#)

[Membership/  
Subscriptions](#)

[FAQ  
Privacy Policy](#)

[Contact Us](#)

[Downloads  
PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



provided to let you put common 'browser' type buttons on your HTML viewing window (back, forward, print, stop refresh and so forth).

- HTML (Edit). In addition to the view-only control, you can do many basic HTML editing tasks with File Explorer as well by using a more advanced control.
- View PDF files. View, print, zoom and find are all available (and more).
- Play sound files in the MP3, Wav, Snd, Au, MID, M3U and MPE formats.
- View video files in the MPEG, MOV, AVI WMV or QT formats.
- View Macromedia Flash animation files.
- Either specify files to display via program code or let the user choose.
- Advanced programmers can register callback functions and/or access all of the methods and properties supported by any of the controls (many more than the template prompts can provide).
- Multiple controls can be placed on the same window, so you could, for example, play audio instructions while the user browses a web file.
- And more...

File Explorer supports Clarion 5.5 only, 32-bit, ABC and Legacy. It is pure template code and does not have any DLLs.

## **Installation**

CapeSoft's installers follow the Clarion 3<sup>rd</sup> Party Association guidelines for install programs (as you'd expect, since they helped to define the standards), and as such there is not a whole lot to say about the File Explorer setup. It located the correct Clarion folders, registered the templates, offered to do the requisite editing of the redirection file, added shortcuts to the documentation to the IDE Accessories/Help menu, and displayed the help file at the end of the install.

Click, click, click, click, click, done. Just the way I like it.

## **Implementation**

In a nutshell, File Explorer is a toolbox full of template wrappers. The main document control template wraps around a variety of specialized OCX/ActiveX/OLE controls from other vendors such as Microsoft, Adobe or Macromedia and provides you with a simplified method of using them in your Clarion programs. You don't have to get down and dirty messing around with the Clarion OLE layer or wade through reams of documentation on obscure properties and methods (unless you *like* doing that sort of thing). For the most part, using File Explorer is a matter of dropping a control template onto a window and then filling out a few prompts.

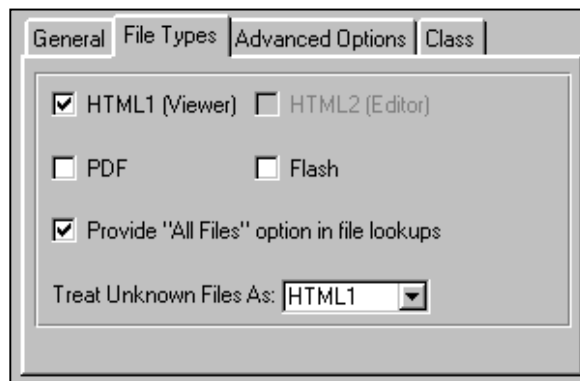
It is important to note that these OCX controls do *not* come with File Explorer. Many are provided either with Windows (the multimedia viewer used for audio and video files, for example) or as part of other common programs (the PDF viewer OCX is installed

automatically when the free Adobe Acrobat Reader program is installed.) The key point is that if, for example, your program displays PDF files, and the machine where you want to run your program does not have the Acrobat Reader installed, that portion of your program will not work.

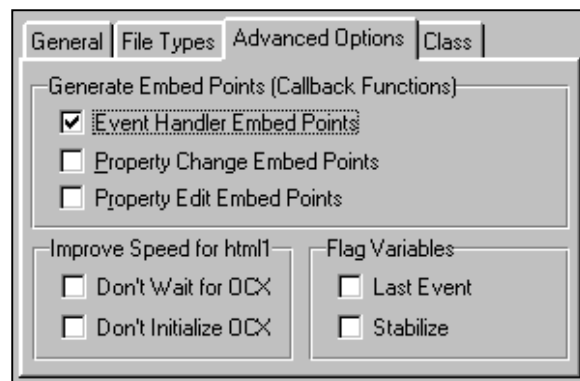
So, a word to the wise: If your program will rely on a particular OCX, it would be smart to either include the tool itself with your own program installs (e.g., the free Acrobat Reader setup program) or provide your users with clear instructions on how to obtain it (e.g., downloading the Flash viewer.)

On to implementing File Explorer...

After you add a global extension, most other File Explorer functionality is as easy as creating a Window procedure and then dropping the viewer control onto it. You then check the box (or boxes) for the types of files you'd like the control to display, maybe set a few more options, and that's about it.

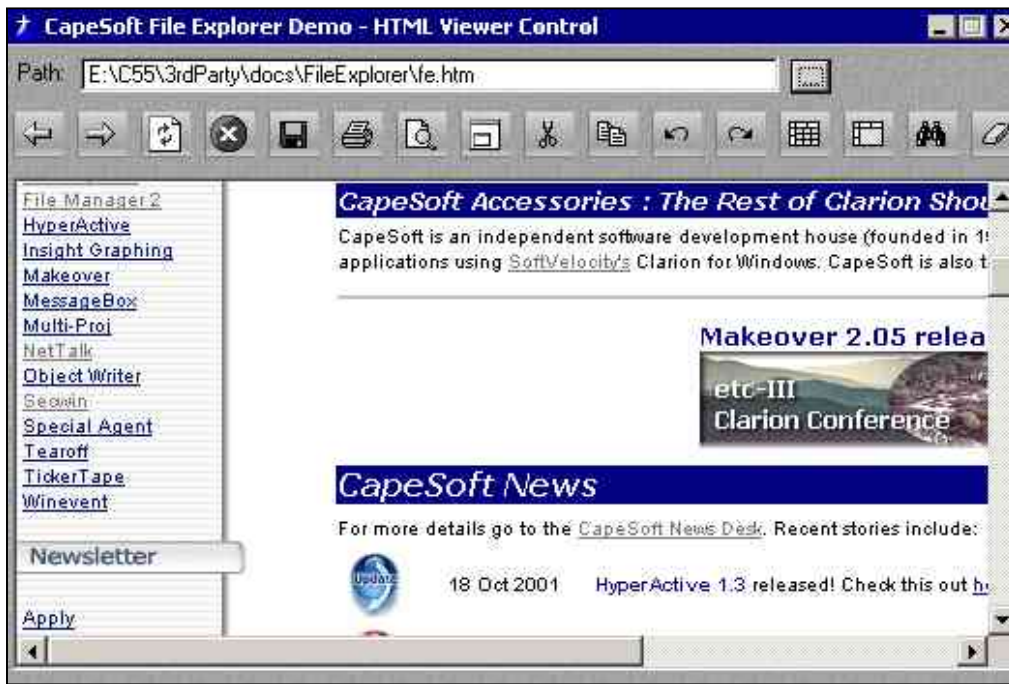


**Figure 1. Selecting the file types a viewer can handle**



**Figure 2. Advanced options for generating embeds or**

If you want buttons to control the viewer (play/pause/stop buttons for audio/video, browser buttons for HTML, etc.), you can just drop on the appropriate File Explorer control template onto the window as well.



**Figure 3. The HTML viewer control in action, browsing a portion of the CapeSoft web site (the file listed at the top was the original local file I loaded)**

Even if you don't place the optional navigation controls on the window, you still get whatever native right-click support the control offers (going back to the previous page in a web browser, for example).

One of my tests was on a Clarion 5.5e, ABC, multi-DLL program. I wanted to use HTML pages to display information about various portions of the program when it was in demo mode. Using the HTML viewer, I found it very easy to load just one local page, and to have all of the other local pages (six in all) linked to each other so that the user could easily read related information. And what if they wanted to print the HTML pages? Easy – I dropped on the File Explorer Print button control template. No muss, no fuss, no embedded code.

Having the HTML pages external to my program meant that it was a snap to edit them on the fly and instantly view the results inside my program. I could also put 'mailto:' links in my HTML pages (these bring up the users email program), and even links to websites out on the Internet.

One tip I'll share here: I found that the viewer in my program usually didn't have enough screen real estate allocated to it to be able to do justice to other web sites (see Figure 3 - most web pages are designed for 800x600 now days). By placing the optional `target="_blank"` parameter on links that left my local pages and went to actual web sites, I caused those pages to pop up in a full-blown Internet Explorer browser. Meanwhile, my own local pages stayed within my program. Here what an external link might look like:

```
<a href=http://www.site.com/page.html
```

```
target="_blank">Go to my site</a>
```

## Performance

I found the performance of File Explorer to be very good, in so far as it can go.

What do I mean by that? Remember, since File Explorer basically just wraps around the other OLE controls, your performance will really be based on two factors: 1) the stability of the C5.5 OLE layer, and 2) the stability and speed of code from Microsoft, Adobe, Macromedia or other vendors. Assuming Clarion 5.5 is stable (knock on wood!), then if another vendor puts out a flaky control there is not a whole lot CapeSoft can do for you.

That said, CapeSoft *has* built-in several options to help "stabilize" some of the more, shall we say, nervous controls such as the Microsoft HTML1 viewer (the read-only one). In reading the documentation and experimenting myself, I was impressed with how some of their workarounds helped make the OCX more stable on various platforms.

The other caveat is that, like most programs, newer versions of the controls tend to work better than older versions and, in some cases, old versions of the controls just don't work at all. CapeSoft does help out by listing the preferred minimum versions in the docs, and which ones work best.

I did run into one major stumbling block in my use of File Explorer. In the course of upgrading from an earlier version to the latest one, the file type for a Print button for my viewer control got out of step with the viewer control itself. With the print button set to `html2` (the edit control) and the viewer itself set to `html1` (the view only control), the app would shift into glacial mode (one small step above totally frozen) whenever I tried to load a document. Since I hadn't changed any template settings myself it took a while to track this down, but setting the two templates back to the same type did fix the problem.

Lest all of that sound too negative, I want to stress that File Explorer does work, and it works amazing well (this from somebody who has spent some very painful times in the OCX trenches in earlier versions of Clarion). And the ability to use the lightweight yet powerful Flash animation format in Clarion programs is (dare I say it)... *way cool*.

## Documentation

Documentation is provided in the form of an HTML file. I've become a major fan of using HTML for documentation – the ability to easily jump from topic to topic, the formatting and graphic capabilities it provides and, last but not least, ease of printing as compared to Windows Help files. By the way, if you want to print the File Explorer document it works out to about 55 pages of solid information.



The docs cover all of the basics: installation, using the templates, examples, FAQ, version history, File Explorer methods and properties, and more. Written in a breezy but informative style, the help is easy to follow and understand.

Three example applications are provided. The first is an extensive demonstration of all the major features and templates (one app for Legacy, one app for ABC), while the other is a simple one-procedure app showing how to display a Flash file full-screen (this app contains exactly one line of embed code). The larger example demonstrates quite nicely that most File Explorer windows can be created with no embedded source code at all.

## Technical Support

CapeSoft's technical support has always been excellent, and did not disappoint this time, either. Questions were fully answered, suggestions were politely received, and support staff were up front in confirming the problem that I mentioned above (and said it would be fixed in the next release.)






The standard support method is email (unless you happen to live in South Africa or like making long distance phone calls.) CapeSoft is also active on the SoftVelocity third-party newsgroup.


## Summary

Since it is impossible to capture how dynamic content such as audio, video or Flash animations perform in a static image in a review, I highly recommend that you download the File Explorer demo from the CapeSoft web site (see below) and check it out yourself.

Could you write code yourself to do everything that File Explorer does? Sure, but if you value your time at anything above a buck an hour, you'd be hard pressed to justify the time and anguish you'd go through versus the very reasonable price CapeSoft charges for File Explorer. You wouldn't get regular free updates and enhancements, either!

Bottom line: File Explorer is a great way to expand the visual vocabulary of your programs, and offers a lot of bang for the buck. If you want to add any kind of pre-formatted or dynamic content to your programs, it is definitely worth checking out.

<b>Overall Product Rating:</b>	
Ability to do the task	
Ease of use	
Ease of Installation	
Documentation	

Technical Support	
Black-Box DLLs/LIBs	No

CapeSoft File Explorer v2.1 costs US\$99, and is available online from <http://www.clarionshop.com>. Clarion versions supported: 5.5 or higher, ABC or Legacy, 32-bit programs only. For more information or to download a free demo of File Explorer, visit the CapeSoft web site at <http://www.capesoft.com>.

---

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

## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

# Clarion MAGAZINE

## Controlling Printers With DevMode (Part 1)

by **Jim Gambon**

Published 2001-10-24

Recently one of my clients made what I thought was a simple request. "Is there any way to save the bin selection for this report?" she asked. "I always have to change it, and it would be nice not to have to remember." "Sure," I told her. "Let me put that feature in the next version."

Now, the feature I was imagining was not just saving her printer's bin selection. I wanted to create a generic solution that saved and restored any printer setting. To accomplish this I ended up exploring the Windows printer driver API. Yet, because of the power of the Clarion language, I was able to avoid actually having to prototype and use calls to the API. Let me share some of what I learned.

### Clarion's PROPPRINTs

When you design a report in Clarion you can specify many things: the paper size, the orientation, the fonts to use, etc. All of these options, with a few exceptions, are device independent. You design the report and you don't really care what printer the customer is using. If, however, you need run-time control of how a report prints (i.e. print resolution, number of copies, etc), or how it looks (i.e. margins), you can change the properties of both the report and the printer. Changing the [properties of the report](#) is nice, and I've done that in previous projects, but what I needed now were the printer run-time properties. For that I turned to "Printer Control Properties" in the Clarion documentation.

The documentation describes various EQUATES for setting the printer, including number of copies (PROPPRINT:Copies), print quality (PROPPRINT:Resolution), duplex (PROPPRINT:Duplex), as well as bin selection. (PROPPRINT:PaperBin). You can even tell the report what printer to use (PROPPRINT:Device).

To easily use printer run-time properties you must first include the PRNPROP.CLW file in your application. You can do this in any data section; I recommend using the Global Data embed area, or the After Global INCLUDEs embed area. Make sure you indent several characters to keep the code from being generated into column one.

#### Search

#### Home

#### COL Archives

#### Advertise

#### Reader Comments

#### Call for Articles

#### Information

[Log In](#)

[Membership/  
Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

#### Downloads

[PDFs](#)

[Freebies](#)

[Open Source](#)

#### Site Index



The `INCLUDE` statement is:

```
INCLUDE( 'PRNPROP.CLW' )
```

Please note that some third-party templates require this `INCLUDE`, but most of them also have a "Do not include `PRNPROP.CLW`" setting to avoid the conflicts of including a file more than once. Clarion 5 and higher added a `ONCE` directive on the `INCLUDE` statement to tell the compiler to only perform that actual `INCLUDE` once no matter how many times it is actually included in the code.

The `PRNPROP.CLW` file is just a list of equates, which are not allocated any memory. These equates are just names to keep you from having to remember all of the property numbers. Setting any of these printer properties is very straightforward. Before the report is opened, you use code to change the settings of the currently selected printer, something like this:

```
! To set three copies:
PRINTER{PROPPRINT:Copies} = 3
! Set image scale to 50%:
PRINTER{PROPPRINT:Scale} = 50
! The PRNPROP.CLW file also includes some helpful
! EQUATES for various settings like paper bin.
! To set the paper bin to "lower tray:"
PRINTER{PROPPRINT:PaperBin} = PAPERBIN:Lower
```

If you plan on doing much work with reports, you should review the `PRNPROP.CLW` include file for all the possible run-time properties.

## The biggest property of them all

If I wanted to design my own printer dialog window, then all of these discrete properties would come in quite handy. However, I wanted to save and restore any printer setting no matter how obscure. For example: my HP laser printer driver described things like "PostScript Output (optimized for speed or size)" and "Paper finish (matte or glossy)." If my user changed these settings I wanted to be able to save and restore these as well.

Luckily, Clarion has a `PROPPRINT` that includes all of the properties of a print driver. This encompassing property is `PROPPRINT:DevMode`, which Clarion's documentation describes as *"the entire device mode (devmode) structure as defined in the Windows Software Development Kit. This provides direct API access to all printer properties. Consult a Windows API manual before using this."* Well, this was somewhat encouraging. I could get to all of the print driver's properties, but I realized I would also have to understand the device mode structure to avoid causing a General Protection Fault on my user's computer.

## Structures and GROUPS

In Clarion a structure is defined as a `GROUP`. This is, as you may guess, a group of variables that you can reference as a single unit.

You can also reference each individual variable within the `GROUP` structure. I had used `GROUPS` in the past, so I was pretty confident I could use the `DevMode` structure that is defined in the Clarion documentation. Unfortunately the Clarion `DevMode` structure is 16 bit only, and the Clarion docs said "*the devmode structure is different in 32-bit (consult a Windows API manual).*" Ok, so it was now time to open the API guides to Windows software development.

My standard (printed) API guide, Dan Appleman's [Visual Basic Programmer's Guide to the Win32 API](#), describes the `DevMode` structure as the group of variables that define the capabilities of a device, be it a display driver or a printer driver. I like the Appleman book because it demonstrates ways to use the lower-level API in programs built in a higher level language. Of course it gives these examples and data structures in terms of Visual Basic, so, to convert the structures into Clarion syntax, I often use the *WIN API Viewer* (`WINAPI.EXE`) program included with Clarion. This has always been particularly helpful because it has most of the prototypes, constants, and structures needed for Windows programming in the Clarion environment.

I will not drag you down the paths I took to come up with a valid `DevMode` structure, but let me give you a word of warning: do *not* use the `DevMode` structure as defined in the *WIN API Viewer* program that ships with Clarion. The primary problem is that many of the fields in the `DevMode` structure are 16 bit `SHORTs`, but the *WIN API Viewer* has them defined as `SIGNED`. This is fine in 16bit programming where `SIGNED` is defined as a 16 bit `SHORT`, but 32 bit `SIGNED` variables become 32 bit `LONGs`, and this throws the structure completely off.

Using other references (primarily [MSDN](#)) I found out that there is a base `DevMode` structure for all Windows versions. Versions of Windows after Windows 3.x have added new fields to the structure as they have added device features. I took the `DevMode` structure from the *Win API Viewer* and modified it to be compatible to the `DevMode` found on MSDN. Listing 1 shows the new Clarion `DevMode`:

### Listing 1. A 16/32 bit DevMode group for Clarion

```

DevMode          GROUP, TYPE
DmDeviceName     BYTE, DIM( 32 )
DmSpecVersion    USHORT ! WORD
DmDriverVersion  USHORT ! WORD
DmSize           USHORT ! WORD
DmDriverExtra    USHORT ! WORD
DmFields         ULONG ! DWORD
DmOrientation    SHORT ! SIGNED
DmPaperSize      SHORT ! SIGNED
DmPaperLength    SHORT ! SIGNED
DmPaperWidth     SHORT ! SIGNED
DmScale          SHORT ! SIGNED
DmCopies         SHORT ! SIGNED
DmDefaultSource  SHORT ! SIGNED
DmPrintQuality   SHORT ! SIGNED

```

```

DmColor          SHORT ! SIGNED
DmDuplex         SHORT ! SIGNED
DmYResolution    SHORT ! SIGNED
DmTTOption       SHORT ! SIGNED
DmCollate        SHORT ! SIGNED
DmFormName       BYTE , DIM( 32 )
DmUnusedPadding  USHORT ! WORD
DmBitsPerPel     ULONG ! DWORD
DmPelsWidth      ULONG ! DWORD
DmPelsHeight     ULONG ! DWORD
DmDisplayFlags   ULONG ! DWORD
DmDisplayFrequency  ULONG ! DWORD
COMPILE( '***' , _WIDTH32_ )
DmICMMethod      ULONG ! DWORD
DmICMIntent      ULONG ! DWORD
DmMediaType      ULONG ! DWORD
DmDitherType     ULONG ! DWORD
dmReserved1     ULONG ! DWORD
dmReserved2     ULONG ! DWORD
! ***

                                END !DevMode Structure

```

Windows 2000 added two other `DWORD` (double word) fields to the end of the structure: `dmPanningWidth` and `dmPanningHeight`. These two fields are specific to display drivers on Windows 2000 only, so I haven't included them in the structure.

Notice I converted all of the field types to the Clarion base types instead of having to rely upon `EQUATE` translation, as defined in the `WINEQU.CLW` include file that ships with Clarion. The `WINEQU.CLW` file has most of the standard equates for working with Windows API prototypes. Another note of warning though: in `WINEQU.CLW` (at least in all Clarion versions up to CW5.5e) a `WORD` is defined as a `SIGNED` and, as I said earlier, a `SIGNED` changes it's size depending on 16-bit or 32-bit compiling. The `DevMode` structure does not need these integer size changes. Instead, it adds fields to the end of the structure depending on compile width (16-bit vs. 32-bit project setting). Also, a `WORD` is always a two byte (16-bit) integer, and therefore should be made equivalent to a `SHORT` or a `USHORT`, not a `SIGNED`.

The above structure is defined as a `GROUP, TYPE`. That means it is just a description of data – no memory is yet allocated. To use this new structure I first had to get it into my application. I could have copied it into a data embed point, but I chose to create a new text file (`DEVMODE.CLW`) and copied the structure into that. Then, in my application, in the same embed that has the `PRNPROP.CLW`, I put the following:

```
INCLUDE( 'DEVMODE.CLW' )
```

This allows me to create new `DevMode` structures where I need them in my application (typically in the Report Procedures). A good place to do this is in the data section for the procedure that needs the `DevMode` structure (usually a Report Procedure). Create a new

GROUP, give it a name like PrnDevMode, and a base type of DEVMODE. This generates code in the procedure's data section similar to this:

```
PrnDevMode          GROUP (DevMode)
                   END !GROUP
```

To assign the printer's device mode to the new variable (i.e. PrnDevMode) you do a simple property assignment in the CODE section of a procedure. The Clarion run-time libraries then takes care of the Windows API functions needed to extract the DevMode structure and place it into your variable. Clarion also makes sure it moves as much data from DevMode into your variable as your variable can hold. Here's the code:

```
PrnDevMode = PRINTER{PROPPRINT:DevMode}
```

## The Data in the Structure

So now that I had the DevMode structure, what could I do with it? As it is, when it comes to printers, there is very little that cannot be done with PROPPRINT properties. In fact some of the fields found in the DevMode are harder to work with than the corresponding PROPPRINTs. For example, the device name is placed into an array of BYTES. In order to get the device name (and, similarly, the form name) into a variable I can use, I had to create new CSTRING variables OVER the DevMode's dmDeviceName and the dmFormName. The OVER attribute tells the compiler that this variable will not use its own memory space, but will share the space used by the variable OVER names. That way I can use the DeviceNameString variable instead of having to work with the BYTE array, and if the BYTE array data changes, the OVERed variable also changes. The declaration looks like this:

```
! In the Local Data Embed after the
! Generated Declarations
DeviceNameString CSTRING(32),OVER(PrnDevMode.dmDevicename)
FormNameString   CSTRING(32),OVER(PrnDevMode.dmFormName)
```

After reviewing the fields available in DevMode, I found they corresponded roughly to the PROPPRINT properties. There are some fields not available as PROPPRINT properties. For example, the current printer's orientation is stored in the dmOrientation field where 1=Portrait and 2=Landscape. To toggle the current printer from portrait to landscape you cannot set a single PROPPRINT property. You must retrieve the DevMode structure, change the dmOrientation field, and then place the DevMode structure back into the printer driver:

```
! Retrieve the base DevMode structure
PrnDevMode = PRINTER{PROPPRINT:DevMode}
IF PrnDevMode.dmOrientation = 1
  ! If currently portrait set to landscape
  PrnDevMode.dmOrientation = 2
! Send DevMode back
```

```
PRINTER{PROPPRINT:DevMode} = PrnDevMode  
END !IF
```

Conversely, there are PROPPRINT properties available in Clarion that have no corresponding DevMode field. PROPPRINT:PrintToFile is an example of this, and Steve Parker covered this topic in a [recent article](#).

So, I could retrieve the DevMode structure, set new values, and send the structure back to the printer driver. Did this give me enough information to save and restore any property that my customer chose? Dan Appleman's API guide told me that it did not. To quote: *"It is important to keep in mind that in addition to the standard fields defined in the listing, each printer driver may define its own private data area. The fields in this private data area are not accessible to the programmer, but it is necessary to allocate sufficient buffer space to include this data area when using the (API) function(s). Some printer drivers will fail if this extra space is not present."*

Uh oh. "private data area...not accessible to the programmer...but...necessary...(or) some printer drivers will fail." These are strong words of warning. In [part two](#) of this article I will describe ways to work with the "private data area" of the printer driver using only Clarion language statements.

---

*After receiving a Mechanical Engineering degree from the Georgia Institute of Technology, [Jim Gambon](#) decided that he enjoyed working with computers too much to actually get a job in the engineering field. He began programming banking and small business applications in Clarion 2.1 and has kept at it ever since. Jim and his wife Nicole own JNData, LLC in Dawsonville, Georgia.*

## Reader Comments

[Add a comment](#)

**API examples of DevMode work are mostly Wrong! The stuff in...**

**Carl, thanks for the MS KB article concerning the actual...**

Copyright © 1999-2001 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.





published by  
**CoveComm Inc.**

# Clarion MAGAZINE

## Clarion Developers Conference 2001 Latin America

Published 2001-10-24

The Clarion Developers Conference 2001 Latin America will be held November 8-10 in Buenos Aires, Argentina. All conference proceedings will be in Spanish only. Accordingly, the following pages are also in Spanish only!

[Conference schedule](#)

[How to get there](#)

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 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.



published by  
CoveComm Inc.

# Clarion MAGAZINE

## The Clarion Advisor: Creating A SOAP Client

by **Brian Staff**

Published 2001-10-26

It is possible to write Clarion code that acts as a SOAP client which sends requests to any SOAP server. My own work recently allowed me to explore this new SOAP technology; I needed to retrieve customer information from a web based ASP (Active Server Page) application which manages a MSSQL database. The returned customer information in my case is XML of one or more customer records, but it could also be just plain text, as demonstrated with the code below.

The SOAP server part which handles the "facade/delegate" is written as a VB COM object, largely because it's compiled and faster than ASP, which is interpreted. I'm using Jim Kane's [COM code](#) to create a SOAP client which talks to the VB SOAP server. Here's a snippet of Clarion code that works with the Microsoft SOAP examples:

```
oClient =& NEW oleTClType
oClient.init('MSSOAP.SoapClient',0)
oClient.CallMethod('ClientProperty |
    ("ServerHttpRequest",True)')
par1 = 'http://localhost/MSSOAPSamples/' |
    & 'Calc/Service/Rpc/AspVbsVb/calc.wsdl'
oClient.CallMethod('mssoapinit |
    ("' & CLIP(par1) & '", "", "" )')
textx = oClient.GetProp('add(124,225)')
```

This code requires that you have the Microsoft [SOAP Toolkit](#) installed. The Clarion code creates a SOAP client which calls one of the SOAP sample services installed on the local machine. This service evaluates the passed math expression ('add(124,225)') and returns the result. Normally a SOAP client and server communicate across a network or across the Internet, of course.

---

*[Brian Staff](#) was born and raised in Rugby, England, and lived for 28 years in Vancouver, Canada. He worked too many years for Honeywell on mainframes, and spent six years as an independent developer, including four years developing*

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



*software for DirecTv. Brian currently develops point of sale systems and web applications for JDA Software in Phoenix, Arizona (it's a dry heat). A member of Team Topspeed since Feb 1996, Brian is also the author of the Xplore templates and is a coach and volunteer web site developer for the local soccer community. He is married to Valerie, has three soccer-playing daughters, and is a former international level rugby referee.*

## **Reader Comments**

[Add a comment](#)

**Glad to see something in a Clarion article about SOAP. As...**

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

# Clarion MAGAZINE

## Large Table Performance in MySQL

by **Dave Harms**

Published 2001-10-26

Recently there's been some discussion of MySQL/MyODBC performance with large tables. As most of the work I do is with tables of less than 10,000 records, I haven't really noticed the problems others have reported. But I decided it was time to do some testing with large data sets.

As it happens I do have one fairly large table, a web server log with over five million records. In order to test this table, however, I first wanted to move it to a test server.

### Moving tables around

There are several ways to move MySQL tables from one server to another. If you're dealing with ISAM or MyISAM tables (these are the MySQL default table types for older and current versions) then you can simply shut down the MySQL server and copy the data files to a new location (see my [previous articles](#) on MySQL for more on table types). On my Linux server, each MySQL database is stored in its own directory under the /var/lib/mysql directory. After copying the data files across I simply restarted MySQL, added permissions as necessary (using the GRANT statement), and started working with the tables.

There is one caveat to copying MySQL tables: if you're using a version of MySQL older than 3.23 then chances are your tables are the older ISAM rather than MyISAM tables. ISAM tables are only portable within platforms, i.e. you can copy MySQL ISAM tables from one SPARC machine to another, or one Intel machine to another, but not from a SPARC to an Intel machine (because of byte ordering differences). MyISAM tables, however, which appeared as of MySQL 3.23, are binary compatible between platforms. If you're using a more recent version of MySQL but began with a pre-MyISAM version then you may need (or want) to convert your tables to MyISAM format. You can do this with the ALTER TABLE statement:

```
ALTER TABLE tablename TYPE=MYISAM;
```

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



**NOTE:** In a [previous series](#) I also discussed the InnoDB table type, which supports transactions and row level locking; you cannot physically copy InnoDB tables around, or at least I wouldn't advise trying it!

If you're not sure what type your tables are now, you can issue the `SHOW TABLE STATUS` command. This will give you many columns of information including the average row length, data size, last autoincrement primary key value and more, but I've truncated the results here to just the first four columns:

```
mysql> show table status;
+-----+-----+-----+-----+
| Name          | Type  | Row_format | Rows  |
+-----+-----+-----+-----+
| AccessHistory | ISAM  | Dynamic    | 1152525 |
| AccessLog     | ISAM  | Dynamic    | 14063  |
+-----+-----+-----+-----+
```

**TIP:** When I get a really wide result back from the `mysql` client, the lines wrap making it difficult to read. I usually paste such results into Windows Notepad and turn off word wrap. The Clarion editor also works fine. I can easily copy and paste because I use a Windows terminal program (such as [CRT](#)) to telnet or SSH to the Linux server.

If you get a syntax error trying to execute `SHOW TABLE STATUS` then you're using an older version of MySQL and you have ISAM tables.

If copying tables across whole isn't an option you can always use the `mysqldump` command line utility. Well, mostly always. I found I ran out of memory when I tried this on my five million record table, but I might have had better success with the `-q` (do not buffer) option. The `mysqldump` utility creates standard output that includes a table creation statement and `INSERT` statements for all of the table's data. You can tell `mysqldump` to dump one table, selected tables, or one or more entire databases. Here are some of the more useful `mysqldump` options:

Option	Description
<code>--add-drop-table</code>	Add a drop table before each create statement.
<code>-A, --all-databases</code>	Dump all the databases.
<code>-a, --all</code>	Include all MySQL-specific create options (see <code>CREATE TABLE</code> syntax).

<code>--allow-keywords</code>	Prefix each column name with the table name to allow keywords as column names.
<code>-c, --complete-insert</code>	Use complete insert statements (with column names) – handy if you are using MySQL dump to move data to a table which has added columns.
<code>--delayed</code>	Insert rows with the <code>INSERT DELAYED</code> command – this assigns a lower priority to the insert so it doesn't interfere with other operations.
<code>-e, --extended-insert</code>	Use the new multiline <code>INSERT</code> syntax. (Gives more compact and faster inserts statements.)
<code>--help</code>	Display a help message and exit.
<code>-F, --flush-logs</code>	Flush the MySQL log file before starting the dump.
<code>-f, --force,</code>	Continue even if a SQL error occurs during a table dump.
<code>-h, --host=..</code>	Specify the server host name. The default host is <code>localhost</code> .
<code>-l, --lock-tables.</code>	Lock all tables before starting the dump. The tables are locked with <code>READ LOCAL</code> to allow concurrent inserts in the case of <code>MyISAM</code> tables.
<code>-t, --no-create-info</code>	Don't write table creation information (the <code>CREATE TABLE</code> statement).
<code>-d, --no-data</code>	Just dump the table structure, not the data.
<code>--opt</code>	Same as <code>--quick --add-drop-table --add-locks --extended-insert --lock-tables</code> . Should give you the fastest possible dump for reading into a MySQL server.
<code>-pyour_pass, --password[=your_pass]</code>	Password to use when connecting – if password is omitted you will be prompted.

<code>-q, --quick</code>	Don't buffer query, dump directly to stdout.
<code>-u user_name, --user=user_name</code>	The MySQL user name to use when connecting to the server. The default value is your Unix login name.
<code>-v, --verbose</code>	Verbose mode. Print out more information on what the program does.
<code>-w, --where='where-condition'</code>	Dump only selected records. Quotes are mandatory, i.e.: <pre>--where=user='jimf' "-wuserid&gt;1" "-wuserid&lt;1"</pre>

Whether you're moving data between servers, doing backups, or just need to export/import data, `mysqldump` is a most useful program. Get familiar with it.

### The test table

For test purposes, I used a modified version of my `DetailHistory` table, a log file that records every request made to the server, whether for an article, an image, or any other file. Here's the relevant table status information for `DetailHistory`:

Name	DetailHistory
Type	MyISAM
Row Format	Dynamic
Rows	5441439
Average Row Length	54
Data Length	295372180
Maximum Data Length	4294967295
Index Length	62568448
Data Free	0
Auto Increment No	5441440
Create Time	2001-10-25 19:25:15
Update Time	2001-10-25 19:28:34
Check Time	2001-10-25 19:29:41
Create Options	pack_keys= 1



Because this table gets a lot of inserts and has a lot of records, I didn't originally have any indexes declared at all. Actually I didn't even have a primary key declared – this isn't an absolute requirement in MySQL, apparently. The more indexes a table has (or the larger the fields being indexed), the slower inserts will be, since each insert has to also update the indexes. If you're doing a big (I mean *really* big) batch insert of data into a table you may find it faster to drop all the indexes, add the data, and then recreate the indexes after all the inserts are done.

In any case, I decided that that a primary key would be a good idea, so I added one:

```
mysql> alter table DetailHistory add index
      > DetailHistoryIdx_ReqDateTime(ReqDateTime);
Query OK, 5441439 rows affected (4 min 19.69 sec)
Records: 5441439 Duplicates: 0 Warnings: 0
```

As you can see it took this server (a Celeron 400 running RedHat 7.1 with two IBM 20 GB drives mirrored on a 3Ware Escalade controller) just over four minutes to add an autoincremented primary key value to a table with 5.5 million rows and almost 300 megs of data.

Judicious use of indexes becomes critical on large data sets. For instance, the DetailHistory had a user field which I suspected was unused, since the code that inserts these log files is completely separate from the authentication system. I decided to have a look:

```
mysql> select user from DetailHistory
      > where user <> '';
+-----+
| user |
+-----+
| testname |
| testname |
| testname |
| testname |
+-----+
4 rows in set (11 min 15.87 sec)
```

Because there was no index on the user field, the server had to look through all five and a half million records, and the query took a substantial length of time. To compare times, I created an index on the user field:

```
mysql> alter table DetailHistory add index
      > DetailHistoryIdx_User(user); Query OK, 5441439 rows
affected (6 min 3.31 sec) Records: 5441439 Duplicates: 0
Warnings: 0
```

I then reran the query:

```
mysql> select user from DetailHistory
      > where user <> '';
+-----+
| user   |
+-----+
| testname |
| testname |
| testname |
| testname |
+-----+
4 rows in set (24.71 sec)
```

**Not that speedy, but better. Of course, looking for a specific user value is quite fast:**

```
mysql> select user from DetailHistory
      > where user = 'testname';
+-----+
| user   |
+-----+
| testname |
| testname |
| testname |
| testname |
+-----+
4 rows in set (0.00 sec)
```

**A limited select in user order is similarly fast:**

```
mysql> select user from DetailHistory
      > order by user desc limit 4;
+-----+
| user   |
+-----+
| testname |
| testname |
| testname |
| testname |
+-----+
4 rows in set (0.01 sec)
```

**So much for testing. I don't actually need that user field, so I dropped it from the table.**

```
mysql> ALTER TABLE DetailHistory drop column user;
Query OK, 5441439 rows affected (4 min 26.71 sec)
Records: 5441439 Duplicates: 0 Warnings: 0
```

**I do want at least one other index on IP address, so I have something other than the primary key for browse testing:**

```
mysql> ALTER TABLE DetailHistory
      > ADD INDEX DetailHistoryIdx_IP(IP);
Query OK, 5441439 rows affected (6 min 30.36 sec)
Records: 5441439 Duplicates: 0 Warnings: 0
```

Again, it's a few minutes to complete the change. Finally, it's time to do some testing!

## Testing a large MySQL table

When it comes to really large MySQL tables and the MyODBC driver, I have bad news, a workaround, and a ray of hope. The bad news is that you really can't use large MySQL tables with Clarion/ABC at present (at least as of 5.5E), because the ODBC driver isn't setting the `LIMIT` clause on the `SELECT` statement. That means that if, for instance, you have over five million records in the table, the server will have to retrieve all of those records before your browse can begin to display! That is, of course, unworkable; I have filed a bug report.

The workaround is to use `PROP:SQL` to specify the `SELECT` statement, with `LIMIT` clause. When you explicitly set the SQL statement and issue a `NEXT()`, you can use `LIMIT` with MySQL tables. I would expect the CCS templates (<http://www.cscowboy.com/products.htm>) would work fine with MySQL. Unfortunately, I haven't found an easy way to circumvent the ABC classes' communication with the driver.

The ray of hope is a bug fix - bug fixes spring eternal!

Another point worth noting is that if you have a filter on your huge file, and that filter uses keys and is therefore reasonably speedy, you can probably still use an ABC browse with MySQL. You'll be retrieving more records than you need to see, but if you have a reasonably fast server and network, you might be able to *temporarily* get away with a select that returns a few hundred or a few thousand records, even though you only view 20 or so at a time. Yes, you'll be putting the server to way too much work, but if you have some capacity to spare this may get you by.

## Summary

The MySQL server is quite capable of handling large files; unfortunately, Clarion ABC applications, at present, don't pass `LIMIT` clauses through to the back end, so if you don't have a filter on your data, any page browse will cause the server to retrieve *all* the records in the table. If that's your situation, you can use `PROP:SQL` in hand code, or you can use a commercial product that uses `PROP:SQL`, or you can wait for a bug fix.

---

*David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of Developing Clarion for Windows Applications, published by SAMS (1995). His most recent book is [JSP, Servlets, and MySQL](#), published by HungryMinds Inc. (2001).*

## Reader Comments

[Add a comment](#)

**[I've found a new candidate for "best MySQL administration..."](#)**

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

**INVEST**  
in your own abilities

**Clarion**  
magazine

published by  
**CoveComm Inc.**

# Clarion MAGAZINE

## The Clarion Magazine Free Sampler

Welcome to the Clarion Magazine Free Sampler. Here you'll find a representative selection of articles published online in past issues of Clarion Magazine, along with the [FAQ](#) and [subscription](#) information.

Clarion Magazine began publication in February 1999, and thanks to the support of Clarion developers in over 75 countries, has grown into the most comprehensive source of Clarion programming information anywhere. As of August 2001, the [www.clarionmag.com](http://www.clarionmag.com) site contained over one million Clarion-related words.

Clarion Magazine publishes more or less continuously (48 weekly issues per year), with articles covering all aspects of Clarion application development, from the basics of using the Application Generator to hard core hand code, and everything in between. Clarion Magazine is more than just a static repository of information, however. [Subscribers](#) can attach comments to articles, search for articles, participate in weekly surveys, and more.

I'm confident you'll find Clarion Magazine an indispensable resource.

If you have any questions not answered by the [FAQ](#) and the [subscription page](#), please contact [subscriptions@clarionmag.com](mailto:subscriptions@clarionmag.com).

Dave Harms

Publisher

### Distributing the PDF

You are free to distribute the [PDF version](#) of the Clarion Magazine free sampler to anyone you like. You may not, however, publish any of these articles online (other than as a link to the PDF), alter the PDF in any way, or charge for the PDF (other than a nominal distribution fee).

If you are reading the PDF version, you can find the most recent list of free articles at:

[www.clarionmag.com/cmaga/freeindex.html](http://www.clarionmag.com/cmaga/freeindex.html)

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information](#)

[Log In](#)

[Membership/  
Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



You will need a [subscription](#) or a [free membership](#) to read most of the following articles online. There is no such restriction on the [PDF](#) version.

### **The Clarion Advisor: Speed up your APP debugging with a PRJ**

Certain kinds of errors are a lot easier to fix if you're not stuck in a modal error editor window. By using a PRJ with your APP, you can get full use of your editor again.

*Posted Monday, February 08, 1999*

### **ABC or Legacy: Which Templates Should You Use?**

Which templates should you use? ABC offers power and complexity; legacy offers comfort and predictability. Do you need to switch to ABC?

*Posted Monday, February 08, 1999*

### **Product Review: Xplore Templates**

Datamatrix's Xplore templates turn browses into your customers' playthings. Sort on any column, reformat, do quick reports, even export to other apps.

*Posted Monday, February 15, 1999*

### **Don't Know, Do Care - A Philosopher Looks At OOP**

Steve Parker muses on the meaning of object-oriented programming as implemented in Clarion.

*Posted Monday, February 15, 1999*

### **The Clarion Advisor: Topspeed Driver Error Codes**

Nigel Hicks has kindly provided a comprehensive list of TPS driver error codes. The Clarion Advisor has added some redirector information and a survey of which errors were reported in the newsgroups, and when.

*Posted Monday, February 22, 1999*

### **The Novice's Corner - Getting A Grip On Clarion**

For those of us who have been using Clarion for years, the application development environment, with all of its idiosyncrasies, is second nature. If you've just picked up Clarion, however, your initial reaction may be more one of confusion than familiarity. This article provides a road map to the Clarion development environment, including the use of templates and the "standard Clarion" approach to application design.

*Posted Monday, February 22, 1999*

### **David Bayliss On The ErrorClass**

David Bayliss begins his comprehensive series on the inner workings of ABC with a discussion of ErrorClass, one of the most fundamental and basic ABC classes.

*Posted Monday, February 22, 1999*

### **The Clarion Magazine FAQ**

Wondering what Clarion Magazine really is? Not sure if you want to subscribe? Chances are the Clarion Magazine FAQ will set your mind at ease.

*Posted Wednesday, February 24, 1999*

### **The Novice's Corner - Understanding Templates And Embeds**

In this second installment of the Novice's Corner Dave Harms reviews the standard categories of templates and explains how the templates make it possible to embed code in your generated application.

*Posted Friday, March 19, 1999*

### **How ABC Handles Multiple Sort Orders**

A need for speed (sorts) takes Steve Parker down a winding path to ABC's handling of multiple sort orders.

*Posted Monday, April 05, 1999*

### **How To Convert Your Database To SQL**

Scott Ferrett has kindly given permission to Clarion Magazine to reprint his Euro DevCon '99 presentation on converting an ISAM (TPS) database to SQL. This is essential reading for anyone considering the move to SQL.

*Posted Monday, May 03, 1999*

### **The Cranky Programmer - Install THIS!**

The Cranky Programmer loves Clarion add-ins. What he doesn't care for is the hack job a lot of third party vendors do on their install programs.

*Posted Tuesday, September 07, 1999*

### **DevCon Details: Welcome And Keynote Address**

In his keynote address, Hank Asher, founder of eData, outlined the future of application service providers and offered a unique service to Clarion developers.

*Posted Wednesday, October 06, 1999*

### **Editorial: DevCon Wrapup**

Dave Harms rolls up his editorial shirtsleeves and reviews TopSpeed's future direction as revealed through DevCon '99.

*Posted Wednesday, October 20, 1999*

### **Stephen Mull's Guide To Converting To MS-SQL**

Stephen Mull has written a detailed account of his transition from TPS files to MS-SQL. This is essential reading for anyone considering SQL. Free access: no subscription required.

*Posted Tuesday, November 02, 1999*

### **Presenting Many-To-Many Relationships**

Many-to-many relationships are a common part of database designs, but they can be tricky to present to your users. Tom Ruby explains three approaches to making M2M work for the end user. Part 1 of 2.

*Posted Tuesday, November 09, 1999*

### **WebBuilder Skeleton Basics: Which? When?**

Steve Parker begins a weekly series explaining the inner workings of Clarion web application skeletons.

*Posted Tuesday, January 11, 2000*

### **The Bitlist class and templates**

Jeff Slarve's open source BitList template and class are a great way to manage large numbers of True/False values with a minimum of overhead.

*Posted Tuesday, March 07, 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.

*Posted Tuesday, April 04, 2000*

### **Interview: SoftVelocity's Bob Zauere**

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

*Posted Tuesday, May 16, 2000*

### **Clarion For Zebras: Working With Barcodes (Part 1)**

Simon Brewer explains the ins and outs of using barcoding technology with Clarion. Part 1 of 2.

*Posted Tuesday, June 20, 2000*

### **Using API Threads - Part 1**

Clarion does an excellent job of managing threads for the developer, but there are times when the standard thread handling just doesn't cut it. Jim Kane shows how to safely use API threads in Clarion.

*Posted Tuesday, July 18, 2000*

### **Five Rules for Managing Complexity: Part 1**

Tom Ruby kicks off a five part series on managing application complexity with a rule about repeating fields.

*Posted Tuesday, August 22, 2000*

### **An Introduction To Writing Templates: Part 1**

Writing templates is easier than you think; John Morter explains his borrow-and-adapt approach to becoming productive quickly. Part 1 of 2.

*Posted Tuesday, October 10, 2000*

### **Clarion Essentials CBT From SoftVelocity**

Tom Hebenstreit reviews SoftVelocity's Clarion Essentials Computer-Based Training CD. As a step up from the original Clarion Foundations CBT, which concentrated mainly on learning the basics of the Clarion language and IDE, the Essentials course tackles more advanced concepts and features of the Clarion IDE and language.

*Posted Friday, December 22, 2000*



### **Sending Clarion Reports as Email Attachments (Part 1)**

The email capability in version 5.5 is a nice addition to the Clarion toolset. What is still missing however, is the ability to easily send a report as an email attachment. In this article David Potter demonstrates one possible solution to this problem. Part 1 of 2.

*Posted Tuesday, January 09, 2001*

### **Introduction To SQL: Part 1**

In the first installment of this new series, Dave Harms compares TPS and SQL databases and explains why SQL is important to Clarion developers.

*Posted Wednesday, February 28, 2001*

### **The Cranky Programmer: To Patch or Not to Patch, That is the Question**

Last week the long awaited first batch of fixes for Clarion 5.5 was actually unleashed upon the world. This week, Clarion Magazine unleashes Cranky upon the unsuspecting patch.

*Posted Tuesday, March 06, 2001*

### **Writing Classes That Create Word Documents**

It's easy to create Word documents from a Clarion application, using Jim Kane's OLE classes. Dave Harms looks at the OLE code, and ponders a set of Word-specific classes.

*Posted Tuesday, April 10, 2001*

### **Frequently Asked Questions**

Do you know how many times per year Clarion Magazine publishes? Ever wonder how many words of Clarion-related information there are on the ClarionMag web site? Do you want to know who's behind Clarion Magazine? Read on...

*Posted Monday, August 20, 2001*

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

# Clarion MAGAZINE

## Controlling Printers With DevMode (Part 2)

by **Jim Gambon**

Published 2001-10-30

[Last week](#) I showed a DevMode structure which you can use in your own applications to manipulate certain printer properties. I also pointed out that some printers may have a "private data area" that may contain settings that aren't defined inside the standard DevMode. For instance, my HP laser printer has features like "PostScript Option" and "Paper Finish." It turns out that each printer driver and each display driver may put their driver specific data at the end of the basic DevMode structure. This extra data can be quite sizable, and this "full" DevMode, which includes the printer's "private data," is what I wanted to be able to completely save and restore for my user.

Please review the DevMode data structure (i.e. the GROUP) that I described in [last week's article](#). The fields that define the size of the full DevMode structure are dmSize (the size of the base data area) and dmDriverExtra (the size of the driver specific data found beyond the dmSize limit). Since this extra data can be different for every printer driver, I knew that I would have to dynamically allocate some memory to store the driver's "full" DevMode in memory. Then, once I had allocated the memory needed to store the full DevMode structure, I would need to retrieve that data into this newly allocated memory.

At first look it seemed like I was going to have to resort to low-level memory allocation and manipulation functions like `malloc()` and `memcpy()`, and then use the Window API `DocumentProperties()` and `DeviceCapabilities()` functions to retrieve the full DevMode.

Turning first to the tools available in the Clarion language I found the `NEW` function. The `NEW` function allows you to allocate the memory for variables at run-time. You need to know the `TYPE` of the variable when you design the program (`STRING`, `GROUP`, `CLASS`, etc), but the memory does not get allocated until you perform the `NEW` statement. The variable you create in your code, at design time, is a "reference" variable. A reference variable is like a placeholder. It names the variable, and it describes the data type,

[Search](#)

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader  
Comments](#)

[Call for  
Articles](#)

[Information](#)

[Log In](#)

[Membership/  
Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)



but it does not yet allocate any memory. You can use the Data dialogs in the Clarion IDE to toggle the "reference" checkbox, or create the reference variable with handcode in the Local Data embed. However you do it, the data section code to create, for example, a reference STRING would look like this:

```
ANewStringVar &STRING
```

In the NEW function you describe the size of the allocation:

```
AnewStringVar &= NEW STRING(60)
```

But the DevMode is a GROUP structure. How was I going to allocate a new DevMode GROUP structure for a full DevMode memory size that had to be calculated at runtime? Once again I turned to the Clarion language. Clarion can treat a GROUP like a STRING. This means that I can perform STRING like operations on a GROUP such as "string slicing" to extract a part of the GROUP, or use LEN() to find the allocated size of the GROUP.

On the other hand, treating a GROUP like a STRING means I can also allocate a new STRING and treat it like a GROUP. So now I had my Clarion language solution for the full DevMode structure. First I created a new STRING reference variable in the data section of my procedure along with a "base" DevMode GROUP:

```
FullDevMode      &STRING
PrnDevMode       GROUP(DEVMODE)
                  END !GROUP
```

Then, in the code section, I created a new string of the required size:

```
! Assign the base DevMode structure
! to a DevMode GROUP
PrnDevMode = PRINTER{PROPPRINT:DevMode}
! Then allocate the FullDevMode data
! for the full size "dmSize + dmDriverExtra"
FullDevMode &= NEW STRING(PrnDevMode.dmSize + |
                          PrnDevMode.dmDriverExtra)
```

To have the FullDevMode STRING contain all of the DevMode data just do the PROPPRINT:DevMode assignment again:

```
FullDevMode = PRINTER{PROPPRINT:DevMode}
```

A word of warning though: dynamically allocating memory with a NEW will cause memory leaks if you do not de-allocate the memory when you're done. You must use the DISPOSE function sometime before the procedure closes:

```
! Test to see if you have dynamically allocated
! the memory, and if so DISPOSE it.
! FullDevMode is not NULL
```

```
IF NOT (FullDevMode &= NULL)
    DISPOSE(FullDevMode)
END !IF
```

**You can make a copy of the FullDevMode STRING in memory (useful for saving original settings) with another reference, NEW, and LEN():**

**In the Data section:**

```
SavedFDM &STRING ! Saved Full DevMode
```

**And in the Code section:**

```
! NEW a STRING the length of
! the existing FullDevMode
SavedFDM &= NEW STRING(LEN(FullDevMode))
! then assign FullDevMode to SavedFDM
SavedFDM = FullDevMode
```

## Restoring the printer device

Using the Clarion language, and a small understanding of the Windows API, I now had the full DevMode data for the current printer, with (I hoped) all of the driver-specific data. I could also allocate a new string to make a copy of the FullDevMode string in memory. I still had to figure out how to restore the printer from a saved full DevMode STRING. I realized that I could not just assign the full DevMode STRING to the current printer since the current printer might not have an "extra data" area that was as big as the saved FullDevMode. Even if the full DevMode data area was the same size, the driver-specific data would probably mean different things for each printer driver.

The technique I came up with is to attempt to set the printer to the device name stored in the dmDeviceName of the DevMode structure (i.e. the first 32 bytes of the FullDevMode STRING.) Once the needed printer is established as the "current" printer I then perform the PROPPRINT:DevMode assignment to restore all of the saved DevMode printer settings to the current printer. The code to do this is as follows:

**In the Data Section create a TempDevMode from the base DEVMODE structure:**

```
TempDevMode GROUP(DEVMODE)
    END !GROUP
```

**Then, in the Code section:**

```
! Assign the current PRINTER to the
! needed driver by "string slicing"
! the name out of FulldevMode:
PRINTER{PROPPRINT:Device} = FullDevMode[ 1 : 32 ]
```

```

! Check to make sure it worked
! and that the data is valid
IF FullDevMode[ 1 : 32 ] = PRINTER{PROPPRINT:Device}
  ! Assign the current base DEVMODE
  ! to TempDevMode
  TempDevMode = PRINTER{PROPPRINT:DevMode}
  ! Test to make sure you have the correct
  ! size of FullDevMode and, if so,
  ! reset the FulldevMode data.
  IF (TempDevMode.dmSize + TempDevMode.dmDriverExtra) |
    <> LEN(FullDevMode)
    MESSAGE('Data Size is not correct')
  ELSE
    PRINTER{PROPPRINT:DevMode} = FullDevMode
  END !IF
END !IF

```

Another way of getting the Device Name from the FullDevMode without string-slicing would be to create various DevMode and DeviceName variables OVER the FullDevMode STRING, like this:

```

OverFullDevMode GROUP(DEVMODE), OVER(FullDevMode)
                END !GROUP
FullDMDevice    CSTRING(32), ←
                OVER(OverFullDevMode.dmDevicename)

```

This way you could reference the base DevMode GROUP fields within the FullDevMode STRING.

## Summary

Clarion, the language and the development environment, has always been attractive for its utility in constructing business applications. It does a good job of shielding us from the low-level mechanics of the Windows API, thus allowing us to focus on solving the database and user interface problems of the day. Still, there are times when only a study of the Windows API will yield answers to pressing problems.

A review of the Windows API showed

## The necessary exception to the Full DevMode rule.

**Jim Kane, in a [newsgroup message](#) a while ago, mentioned a printer driver that contained 6K worth of dmDriverExtra data. I thought "wow, this printer must slice bread and fold fitted sheets to have this much driver specific data." The printer driver he was talking about was the Xerox DocuPrint 96 NPS PCL5e printer. The printer almost does slice bread. It duplexes, it binds, it collates, it even drills holes.**

**Unfortunately for my testing, the driver I downloaded from Xerox only had only 321 bytes of extra driver data. It turns out that Xerox has changed the driver. It now stores most of its specific data in a separate file. The name of this**

me ways to retrieve and modify printer settings. However, it was the powerful capabilities of the Clarion language, and its `PROPPRINT:DevMode` property, that allowed me to implement this functionality without having to perform actual Windows API calls. I could retrieve the "full" `DevMode`

**separate file is about the only thing stored in the drivers extra data area that `PROPPRINT:DevMode` returns. To fully restore all of the printer settings, I would have to find that extra file and make a copy of it. I decided against this additional work at this point because it would be a very driver-specific solution, and my customer does not have one of these Xerox monsters (yet).**

structure for a particular printer. I could save that data in memory using a newly allocated string the size of the full `DevMode`. And I had the technique to restore the settings of a particular printer by setting the current printer and then setting that printer's full `DevMode`. All of my testing to this point proved that this "full" `DevMode`, with the `dmDriverExtra` data, did indeed contain all of the settings for each printer driver.

Persistent storage of this "full" `DevMode` string can be accomplished via a `BLOB` or a `MEMO` field in a data table. Since the size of the allocated "full" `DevMode` string is arbitrary, you should also record the stored size in a separate field. For my user I came up with a way of storing the binary "full" `DevMode` structure as text in their personal INI file. How I accomplished that is beyond the scope of this article, but it might be a story for another day.

[Download the source code](#)

**NOTE:** The downloadable code contains an ABC compliant class, and a sample application created in Clarion 5, that demonstrate the techniques described in this series of articles. The class can be used in your own programs. It contains methods for retrieving and modifying most printer settings. The sample application also allows you to explore the structure of the base and full `DevMode` for the printers on your system. The application also demonstrates overriding the class `Save` and `Restore` methods. This might prove helpful in creating a permanent storage mechanism for the printer's full `DevMode` string/group in your programs.

---

*After receiving a Mechanical Engineering degree from the Georgia Institute of Technology, [Jim Gambon](#) decided that he enjoyed working with computers too much to actually get a job in the engineering field. He began programming banking and small business applications in Clarion 2.1 and has kept at it ever since. Jim and his wife Nicole own JNData, LLC in Dawsonville, Georgia.*

## Reader Comments

[Add a comment](#)

Copyright © 1999-2001 by [CoveComm Inc.](#) All Rights Reserved. Reproduction in any form without the express written consent of CoveComm Inc., except as described in the [subscription agreement](#), is prohibited.

# Clarion MAGAZINE

[Search](#)

## Optimizing DLL Loading - Introduction to Rebasing

by **Carl Barnes**

Published 2001-10-31

I frequently see Clarion users on the SoftVelocity newsgroups ask "How do I get my application to load faster?" There is even a topic in the Clarion help entitled "How to Make Your Applications Load Faster", but it fails to mention the best solution provided (using the `LOADONCALL` directive in the export file) does not work for 32 bit applications. In fact, DLLs created with the default Clarion environment have a small omission which causes them to load slower and basically throw a wrench into the gears of Windows memory management. In this article I will show you an easy technique to make your DLLs avoid unnecessary memory thrashing, and which takes just minutes to implement.

If your program is composed of multiple DLLs please take my word that this series of articles is a must read. It is even more important if you have a web or ClarionNET app running under the Application Broker, or applications running via Windows Terminal Server. In these thin-client environments you will have many instances of your application running simultaneously on the server, and so the memory usage benefits and penalties are multiplied.

### Rebasing

One of the more important benefits in multi-user server environments is DLL sharing to reduce memory requirements. The default Clarion project builds a DLL that is either *not* shared, or very poorly shared. The solution is *rebasing*, which is specifying the base address at which your DLL will load. Once rebased your application will load faster, run faster, share DLLs, consume less resources and be easier to debug.

The "small omission" I referred to in the opening paragraph is the setting of the Image Base or preferred loading address of a DLL, when the DLL is created by the linker. This is not a shortcoming of Clarion specifically; all linkers write executable files using a default base address. It is always the responsibility of the developer (that's you) to explicitly define a good base address. Clarion's only shortcoming in this area is in not documenting the exact syntax of

[Home](#)

[COL Archives](#)

[Advertise](#)

[Reader Comments](#)

[Call for Articles](#)

[Information](#)

[Log In](#)

[Membership/](#)

[Subscriptions](#)

[FAQ](#)

[Privacy Policy](#)

[Contact Us](#)

[Downloads](#)

[PDFs](#)

[Freebies](#)

[Open Source](#)

[Site Index](#)





the Image\_Base entry in the EXP file. But the linker provided enough informative error messages that, combined with other information, I was able to nail down the syntax.

Let me emphasize from the start that there is no risk that you could pick a wrong Image Base and create a problem. If you pick an illegal address the linker will spot it, display an error at compile time and revert back to the default. If you pick an address used by another DLL, at runtime the Windows application loader spots it and will rebase the DLL exactly like it does now.

If you want to fully understand every facet of what the Windows loader does to load a new process, you will have a sizable amount of information to digest. I will not dig too deeply into those details in this article; at the end are links to articles and books with more information on the topic. If at any time you get bored by the detailed discussion just skip ahead to the implementation details. While the details to describe the solution are long, you do not need to understand all of them to implement it. You only need to add a single line of code to each DLL to implement the fix, and in Part 2 I'll supply a small template that will make it even easier.

## **A short history of rebasing**

When Microsoft engineers were designing their new 32-bit operating system (Windows NT, not 95) they completely threw out the 16-bit model where applications shared memory and had to cooperate. Too often shared memory turned into the cold-war strategy of "mutually assured destruction." Microsoft started with a clean sheet of paper and designed a preemptive O/S where processes would be isolated from each other. (Besides lots of clean paper, they had the specs to Unix and VAX too.) They also redesigned the file structure and memory structure of EXE and DLL files to make them easier and faster for Windows to load and manage. This structure is named the Portable Executable format or PE. There is a link to more information on PE at the end of this article.

It is important to understand that Win32 uses a *virtual* memory management system (VMM) where each process runs in its own virtual address space. An instance of an EXE, its DLLs, OCXs and data loaded together in memory and running is called a "process." *Every* process running thinks it has 2GB of memory and can address its own private flat memory address space from 0h to 7FFF, FFFFh. Windows maps the virtual requests to real memory as required. A process does not have the privilege level to touch real memory. The point is that even though the application has virtual memory you should just think of it as real memory and never be concerned about other processes running, because you cannot interfere with them and they cannot touch your memory.

Every EXE is normally loaded at address 0040,0000h or the 4MB point in memory. In any Clarion program if you displayed the ADDRESS ( ) of the first procedure in the first module of your EXE

(typically `ADDRESS(main)`) you would get a value of about 4 million for every EXE running. How can every EXE be loaded at the exact same address? Easy - it's a virtual address.

DLLs also have to be loaded into memory, and they are also loaded into the EXEs 2GB address space. If they were not in the same address space then the EXE could not access the DLL code or data.

A DLL has a preferred load address or "image base". The linker uses this image base when it writes the DLL file to "hard code" many 32-bit memory addresses into the object code; for example, references to global and static data. This allows the code to access the data quickly and easily. As an example, if you wrote Clarion code like `GlobalRequest=AddRecord`, the final object code written by the linker might be `mov[0x04040F8],1` where `0x04040F8` is the address of `GlobalRequest` in the 32-bit flat memory address space.

For the Clarion linker the default base address for DLLs is `0040,0000h`, exactly the same as an EXE. How can the EXE and the DLLs all load at the same `0040,0000h` address? They can't; the Windows loader spots the conflict and finds some empty address space (pretty easy in 2GB of space), loads the DLL at that address and performs an address relocation (or rebase) on the DLL. The loader uses a relocation table written by the linker in the DLL header and changes all the references to hard coded addresses in the object code to be based on the actual load address. In the above `mov[0x04040F8],1` example the code page containing the address `0x04040F8` would have to be loaded and modified with the actual rebased address. While this was designed to be a fast process it is most undesirable. Articles on MSDN estimate this rebasing adds 10% to 100% to your load time.

### **Please step out to my garage for an example**

The entire Windows DLL loading scheme can be related to a pegboard out in the garage, loaded with tools of various sizes. Every process in Windows gets its own clean virtual pegboard (your 2GB of address space). My pegboard is private to me and never has to care what other processes are doing on their pegboards. On my pegboard the Windows loader will hang my tools (EXE, DLLs, OCXs). You just can't hang things anywhere; the board has about 32,000 holes for hangers (modules must start on 64k alignment). And you lose a few holes at the corners for screwing the board to the wall (reserved address ranges).



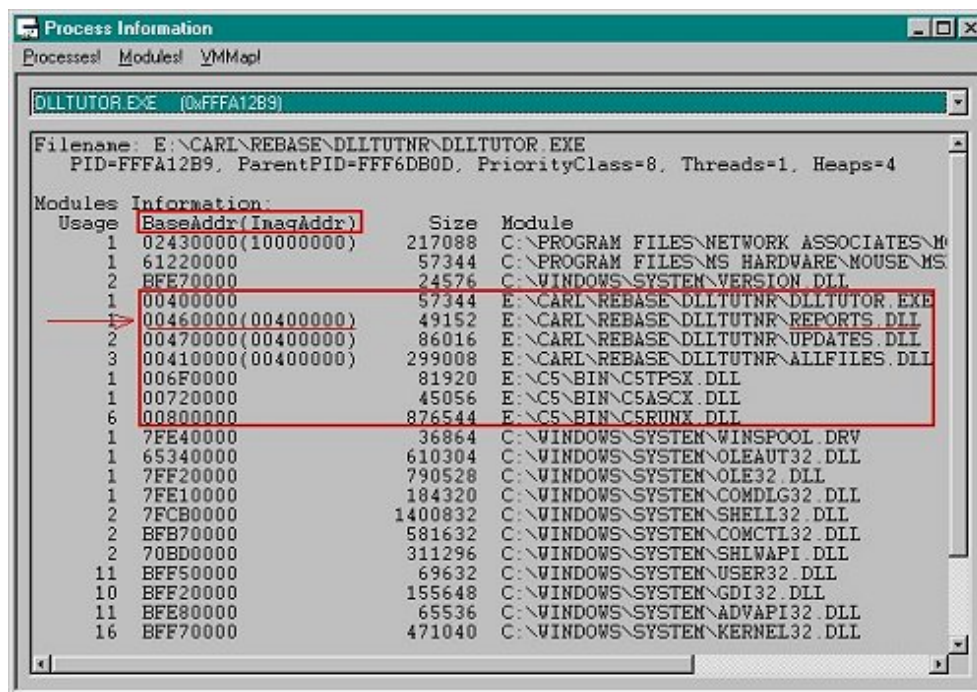
**Figure 1. A tool pegboard (courtesy of [www.plansnow.com](http://www.plansnow.com))**

By default every Clarion EXE and DLL tries to load at the first hole available on the pegboard (address 40,0000h). The EXE is the first to load and always gets the first hole. When the loader tries to hang the first DLL it takes the desired hole number (image base address specified in the DLL header) and checks the board to if the hole is open and the tool will fit in the space available. If you have based your DLL properly the loader can simply hang the tool and move on to the next one.

If the hole is already taken or the tool will not fit in the space, the loader will have to find a hole and space where the tool will fit and hang it there. While this is simple on a pegboard, it cannot be done simply by the loader; the DLL has to be rebased and that starts a cascade of extra work. Picture a first class pegboard where every tool has its silhouette painted behind it. Rebasing is like the loader having to take the time to paint the silhouette behind the tool before it hangs it.

### **An example of a loaded process**

The long description above will be come clearer with an example. Below is a screen shot from a process information utility showing how the DllTutor.App example EXE and DLLs are loaded into memory.



**Figure 2. Modules in the DLLTutor Process**

This utility shows all of the DLLs (also called modules) loaded and the base address at which they were loaded. The address in parentheses next to the base address is the original image base address written by the linker. For example, you'll see that the Reports.DLL is loaded at address 0046,0000h, and the original image address is shown as (0040,0000h), the default address for the Clarion linker. You'll also see that all the DLLTutor DLLs had to be rebased to an available address. The C5 DLLs were loaded at their preferred address because the TopSpeed Development Center carefully assigned each DLL a unique address at link time.

## The Process and Penalties of Rebasing

During a rebase the Windows loader must load the executable code pages and change addresses sprinkled through out your code to be based on the actual loaded address. The linker writes a relocation table into the PE file header that allows this to be an efficient process, but it still is a very undesirable event that can be easily avoided. In the small example DLLTutor app the three DLLs had a total of 9,000 address references that required relocation! The penalties and other negative side effects are as follows:

1. Almost the entire DLL file must be read from disk into memory, whereas for a properly based DLL the loader can read just the PE header sections and initialization code pages.
2. The code pages must be read into real memory. If memory is short other pages will be swapped out to the paging file or discarded. This swapping causes your DLL to load slower and will slow the other processes running whose pages were swapped out since they may eventually need to be swapped back into memory.
3. Scanning the relocation table and relocating the addresses

takes CPU time.

4. The modified code pages must be saved into the system page pool; this takes time, thrashes the disk and takes paging pool space from other programs. This does not happen at load time but later when the system needs real memory and has to swap your rebased pages to the paging file. This slows the performance of other system applications. A properly based DLL can have unneeded pages simply discarded and when the pages are required again they can be reloaded from the disk file.
5. The changed pages are private to your process. This means a rebased DLL's code pages cannot be shared with other processes that use the same DLL. This severely impacts Broker and ClarioNet apps where the app is running multiple times on the server. To repeat, if you do not rebase your DLLs your DLL code is *not* shared between multiple instances of your app running.

There's no doubt that Clarion's default DLL addressing scheme causes some significant performance penalties when you're using multi-DLL applications. Next week I'll show you how to rebase your DLLs so that they'll load faster, and can be shared by multiple instances of a running program. In the meantime, you may want to read through some of the resources listed below.

### **For Further Information**

[Peering Inside the PE: A Tour of the Win32 Portable Executable File Format](#) - A complete and exhaustive look at what's inside an EXE or DLL. Explains the complete relocation process. Includes a utility PEDump that lets you view the relocation data.

[Rebasing DLLs the whole story](#) - This article discusses rebasing and several other factors affecting the load speed of your DLL app. Finding the DLL in the search path is the major delay. Another conclusion is the load time for a DLL is basically fixed so fewer large DLLs is better than many small DLLs.

[Optimizing DLL Load Time Performance](#) - Load optimization and rebasing study by MSDN Magazine Under the Hood column writer Matt Pietrek.

John Robbins. *Debugging Applications* (Microsoft Press), ISBN 0-7356-0886-5. Information on how to create apps that are easier to debug, how to debug at the assembler level and use the address information in the Map file and Dr. Watson logs. CD includes the Windows SDK.

Jeffery Richter. *Programming Applications for Microsoft Windows* (Microsoft Press), ISBN 1-57231-996-8. This book is a "must have" for any serious developer that needs to deal with the Windows API. It covers low level operating system objects like processes, jobs, threads, DLLs and memory.

*[Carl Barnes](#) is an independent consultant working in the Chicago area. He has been using Clarion since 1990, is a member of Team TopSpeed and a TopSpeed Certified Support Professional. He is the author of the Clarion utilities CW Assistant and Clarion Source Search.*

## **Reader Comments**

[Add a comment](#)

**[The Richter book I like is a real bargin now at \\$39 \(retail...](#)**

Copyright © 1999-2001 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.