

# Under Construction: What's New?

by Bob Swart

Two months ago, in *Under Construction*, we talked about Delphi for Workgroups and Component Management. The column focussed on using and sharing a component library on the network. This month I'd like to share some experiences with this technique from the last couple of months...

A shared component library has definite advantages: people share their components with each other, plus enhancements and bug-fixes are automatically made available to all component users.

## Problems

Unfortunately, we've found some difficulties too. One was the fact that BOLIB.DCL (the name of our shared component library) was actively shared by a *dozen* Delphi programmers. This meant that the Component Manager (me) was not able to do updates on BOLIB.DCL unless everyone stopped working with Delphi for a few minutes (otherwise the BOLIB.DCL would be in use and hence locked). Asking someone to stop working with Delphi, if only for a few minutes, is not something you want to do often (not if you want to have friends left at the end of the day...). Moreover, a dozen people being unproductive for even 5 minutes is a total time of one hour. Eight such instances and you have just thrown away a whole man-day of productivity! *[Keep this up and Bolesian will make you Finance Director! Editor]*. Surely there must be another solution...

Of course, the initial idea was that there would only be infrequent fixes and updates. However, over two months we had 3 bug fixes, 4 new components and 3 enhancements to existing components. That's about one update each week. Instead of bugging the BOLIB.DCL users I decided to try and update the library when everybody was at home: either very early in the morning or late at night. In practice, this was not easy to do,

since most of our Delphi users tend to be working while I'm at the office too! While I could find out whether or not BOLIB.DCL was in use (the file was locked), I could not really see *who* was using it.

## What's New?

Apart from the update problem, we faced another problem in the first few weeks: every update, bug fix or enhancement had to be reported to the users, to make sure everybody knew about it. We could have used e-mail or memos, but I decided to let BOLIB.DCL itself tell people if it was new.

During one of the updates, I incorporated a *WhatsNew IDE Expert*. The purpose of this expert is to lie dormant and show the library users if there's something new. While dormant, it looks at a special file on the network (about once every minute). If the date/time stamp of this file has changed since the last time recorded in DELPHI.INI, then something new has happened and the user must be notified. In that case, a message is shown that the *What's New* file on the network has been updated and can be viewed (in a memo on the *What's New* form). The source code for the *What's New Expert* is on the disk

with this issue - it's all fairly straightforward.

## Who's Using BOLIB?

With the *What's New Expert* being part of BOLIB.DCL and active at all times, I could of course perform some more actions. Like registering the time the expert is active (which is equal to the loading and uploading time of the BOLIB.DCL). In fact, combined with the network logon name of the users, I could now maintain a list of people who are running Delphi with BOLIB.DCL (unless they crashed, after which they'd probably restart Delphi and BOLIB.DCL anyway). The *What's New Expert* was extended with this new functionality (see Listing 1).

The initialization code (executed when BOLIB.DCL is loaded) makes sure a message is appended to the logfile and a new *ExitProc* is installed that will append another message to the logfile once BOLIB.DCL is unloaded.

One remaining question was how to get the name of the user? I could use some NetWare or Windows APIs to try to get it, but fortunately, after someone is logged in to our network, an environment string is set to the actual user name: `USRNAME=BOB` in my case, so I

### ► Listing 1

```
procedure AppendMessage(Const Str: String);
var F: Text;
begin
  {$I-}
  System.Assign(F,'Q:\TEMP\BOLIB.LOG'); { a place everyone can write to }
  System.Reset(F);
  if IOResult = 0 then System.Append(F)
  else System.Rewrite(F);
  if IOResult = 0 then writeln(F,Str);
  if IOResult <> 0 then { clear error };
  System.Close(F);
  if IOResult <> 0 then { clear error }
  {$I+}
end {AppendMessage};
procedure ExitProc; far;
begin
  AppendMessage(DateTimeToStr(NOW)+' : BOLIB.DCL unloaded by '+UserName);
end;
initialization
  AppendMessage(DateTimeToStr(NOW)+' : BOLIB.DCL loaded by '+UserName);
  AddExitProc(ExitProc);
end.
```



➤ Figure 1

can use this to get the user names as shown in Listing 2.

The BOLOG support application for the Component Manager does nothing more than scan the logfile on the network, giving me a continuous update of people who have BOLIB.DCL loaded (Figure 1).

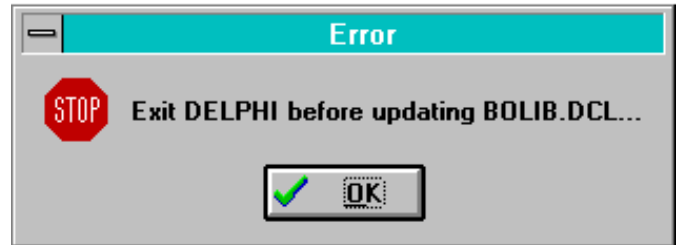
Having this list of current Delphi users, I could finally see when would be a good time to make modifications to BOLIB.DCL. However, during the entire day people were pretty busy using Delphi (as can be seen in Figure 1).

### Local Shared

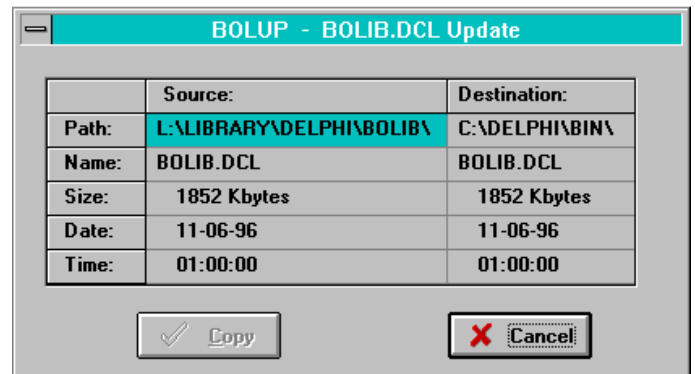
So, a few weeks ago, we decided to switch from the network version of BOLIB.DCL to a shared but local version: shared because it is updated when you logon or by use of a special program, local because it resides on each user's local hard disk.

A special network user group was created with the Delphi users as members. Next, their login script was modified to make sure that BOLIB.DCL would be copied from the network to the local disk if the local version was older. This will automatically update the local BOLIB.DCL every time a user logs on to the network. However, a one-time delay results in a time saving

➤ Figure 2



➤ Figure 3



```
function UserName: String;
var Str: String;
    P: PChar;
begin
    Result := 'USER';
    P := GetDosEnvironment; { Win API }
    while P^ <> #0 do begin
        Str := StrPas(P);
        if Pos('USERNAME=',Str) = 1 then begin
            Delete(Str,1,8);
            Result := Str;
        end;
        Inc(P, StrLen(P)+1) { Fast Jump to Next Environment Variable }
    end
end {UserName};
```

➤ Listing 2

```
program Bolup;
uses
    WinTypes, WinProcs, Dialogs, Forms,
    Bolibupd in 'BOLIBUPD.PAS' {BoLub};
{$R *.RES}
begin
    if (FindWindow('TApplication', 'Delphi') = 0) OR
        (FindWindow('TPropertyInspector', nil) = 0) OR
        (FindWindow('TAppBuilder', nil) = 0) then begin
        Application.CreateForm(TBoLub, BoLub);
        Application.Run;
    end else
        MessageDlg('Exit DELPHI before updating BOLIB.DCL...',
            mtError, [mbOK], 0)
end.
```

➤ Listing 3

when starting Delphi, since loading a local 2Mb BOLIB.DCL is of course much faster than loading it over the network. Also, updates are much quicker, since I can update BOLIB.DCL anytime during the day and once the changes are done, I only have to update the *What's New* file to initiate the *What's New Expert* and within a minute everyone

knows what has been changed in the library.

Another benefit is the fact that Delphi programmers can now test their components before giving them to the Component Manager. Using a shared Component Library on the network gives nobody the chance to add or modify an existing component (apart from the

Component Manager, but in the past he could only make changes if everybody else has gone home). Having a local Component Library means that people can indeed add or modify components.

### **BolUp**

The final support utility that was requested was a special program to copy the network version of BOLIB.DCL from L:\LIBRARY\DELPHI to the local directory C:\DELPHI\BIN. When BOLIB.DCL is updated, this program saves the time taken to logoff and re-logon to the network (to force downloading of the new version of the library). Of course we can't copy the network version of BOLIB.DCL over our local version if Delphi is still up and running using the local version (Figure 2).

To detect if Delphi is running, by the way, can be done as shown in Listing 3 in your main project file. We look for three kinds of windows: TApplication, TPropertyInspector and TAppBuilder, and only if none exists do we run the applica-

tion. BOLUP in action can be seen in Figure 3.

### **Satisfied?**

So, is everybody happy? Almost! We've got still some problems to solve, such as people from the DELPHI group logging on the network from home (with the chance that a 2Mb BOLIB.DCL update is copied to their hard disk - even with a 28.8K modem this could take up to 15 minutes!). We're working on that one...

---

Bob Swart (aka Dr.Bob, <http://www.pi.net/~drbob/>) is a professional software developer using Delphi and C++ for Bolesian, and a free-lance technical author for The Delphi Magazine. For more information about writing your own Delphi IDE Experts, check out *The Revolutionary Guide to Delphi 2*, published by WROX Press. In his spare time, Bob likes to watch video tapes of Star Trek Voyager with his 2-year old son Erik Mark Pascal.