

## A

---

ACT.ZIP on the BBS for downloading by customers with our regular Crystal Reports who want to access ACT! This file unzips to contain

PDBACT.DLL  
CRW.ACT  
PDIRACT.DLL

These files should be copied into the customer's CRW directory and then it will work seamlessly with ACT!.

With the purchase of ACT, the user is told to select CRW.ACT when creating a new report, because it would link four main Act files for use. This is correct. If the user has a special ACT! build (not the usual 2.0, VB, or Pro), there is a file CRW.ACT which is copied to his disk. He/she selects this first when creating a new report. (As an aside, there is also a special PDIRACT.DLL copied to his disk, but he doesn't need to know this.)

---

### Alignment problem

If a field is not reflecting the formatting (ie alignment, border, colors) check if it is formatted to print on multiple lines. If so it will not have the option to change alignment and does not support some other formatting. This option should have been greyed out BUT was not.

---

### Access - Secured Database

An error occurs when Crystal Reports tries to open a Secured Access Database, see Access Online help for a complete description. It states something to the effect that the table location is not correct. The solution is to copy from the MSACCESS.INI file the following:

```
[Options]  
SystemDB=T:\ACCESS\SYSTEM.MDA
```

into either VB.INI or CRW.INI.

## **B**

---

### Btrieve DDFs

In version 1.1 if the size of the combined DDFs (File, Field, Index) were larger than 64K then there would be major problems (including GPFs). The client will have to upgrade to 2.0. DDF's are handled differently in the new version.

---

### Btrieve error - "Compression buffer too short"

Check whether these DDFs can be opened with another tool (Xtrieve?). If they can check the Btrieve and Brequest options in the WIN.INI file. If not the data files do not match the DDFs.

## C

---

### Compiled Reports

As you know when you place a record selection formula based on a partial field within a compiled report it will not allow you to change this at runtime. The work around it to insert a formula that contains the partial field and refer to this formula in the record selection formula.

Name:            @Formula1

Formula:         {file.lastname}[1]

ResSelect:       {@Formula1} = "A"

### Clipper Indexes

Clipper indexes will not be recognized in the current version of Crystal Reports for Visual Basic(2.0)(Sept 3, 1993). Crystal Reports 2.0 Pro will recognize these indexes. Therefore, Jet does not recognize these but PDBXBSE.DLL will. Using the \*.INF file format will not alleviate this either.

## D

---

### Database DDLs

In CRW 1.0, 1.1, 2.0 we shipped exactly 5 database DLLs:

- PDBPDX.DLL - handles all Paradox type tables
  - calls the Paradox engine from Borland, which we ship
- PDBXBSE - handles all xBase database types, such as dBase III, dBase IV, FoxPro, CLipper
  - is self-contained - doesn't call any one else's DLLs, (and that's why it's one of our biggest)
- PDCTBTRV.DLL
- PDBBTRV.DLL - together these 2 DLLs handle bTrieve (non-SQL) databases
  - calls WBTRCALL.DLL and WBTRVDEF.DLL from Novell, which we ship
- PDIROV.DLL - for Borland ObjectVision - not highly used any more, since Visual Basic won the "visual programming" war

In CRW 2.0 PRO we wrote 5 more of our own DLLs, all to handle SQL:

- PDSODBC.DLL - one DLL for any ODBC driver (Microsoft's emerging ODBC standard, Open Database Connectivity)
  - calls MS's ODBC.DLL (we ship this and ODBCINST.DLL) which in turn calls any ODBC driver that the user has on disk - the two we are shipping on disk are SQLSRVR.DLL (for MS SQLServer) and SQORA.DLL (for Oracle)

4 others which all use Q+E library from Pioneer:

- PDSBYBAS.DLL - for Sybase or MS SQL Server (they are identical, just differently marketed) databases
- PDSORACL.DLL - for Oracle
- PDSNETW.DLL - for Novell Netware SQL (it's bTrieve underneath, but you don't need to know that)
  - PDSGUPTA.DLL - for Gupta SQL Base

(In the VB 3.0 bundle we also shipped 3 DLLs PDIRJET.DLL, PDCTJET.DLL, PDBJET.DLL to handle MS Access. In other bundles out there you will find DLLs such as PDIRACT, for ACT! databases, PDSRBASE for Microrim R:Base etc.)

Note 1: About naming conventions: ALL of the database DLLs that Crystal has developed have the names PD\*.DLL - useful to know when helping user's search their hard disks.

Some start with PDB (physical database), some with PDCT (physical dictionary), some with PDIR (physical directory), and some as PDS (physical database SQL). You do not need to worry about this terminology.

Note 2: So there are two different ways to connect to SQL Server or Oracle - via PDSODBC which calls ODBC.DLL etc., or via PDSSYBASE which calls QELIB.DLL etc.

Why 2 ? Well Pioneer's Q+E library was the first WIndows SQL library out there, and is relatively bug-free (you know there are some) so it was the first we've developed for.

Microsoft's ODBC is newer but will no doubt become the most dominant in the industry, but for now it is best for Crystal to support both. As you've seen different drivers have different bugs & features. These drivers should be maturing a lot over the next 8 - 12 months.

Note 3: (and to answer your question) what database formats can a user convert his report to ? Basically the user can convert to any database AS LONG AS IT USES THE SAME PD\*.DLL.

So what this means is if a user has a report using a dBASE III file, and he wants to make the report also work with a dBASE IV (or Fox etc.) file, he can simply choose Database/Set Location... and point at the new file.

Then Verify Database will kick in and it will ask him if we can go ahead and fix up his report which we do painlessly. This works beacuse these tables all use PDBXBSE.DLL. (You shouldn't just trust me by the way, you should try things like this.)

Likewise the user can convert any report using one ODBC driver to any other ODBC driver. But the user cannot do what you asked and that is convert a report using a Q+E driver (e.g. going through our PDSSYBAS.DLL) to one using ODBC (going through PDSODBC.DLL)

Note 4: This is a shortcoming that we hope to address in CRW 3.0 - in even more general cases we have heard of users building a report on Paradox data and wanting it to work on a SQL database. There are quite a few complications in doing this that I won't go into.

..... For now the user does not need to throw away all of his report - but those database fields anyway (he can keep his text, titles, etc. delete the old table, add the new one and reinsert fields; his formulas he can just edit)

---

Using the Date(yyyy, mm, dd) function to convert a string to a date.

If the string field appears as "1993, 09, 27".

@ToDate

Date ({file.field}[1 to 4], {file.field}[6 to 7], {file.field}[9 to 10]);

If the string field appears as "93-09-27".

@ToDate

Date ("19" + {file.field}[1 to 2], {file.field}[4 to 5], {file.field}[7 to 8]);

If the string field appears as "09/27/93".

@ToDate

Date ("19" + {file.field}[7 to 8], {file.field}[1 to 2], {file.field}[4 to 5]);

If there aren't leading zeros then things get a bit complicated for dates like 1/1/93, and 12/30/93 both to work.

```
@ToDate
```

```
if
```

```
    Length ({file.field}) < 8
```

```
then
```

```
    if
```

```
        Length ({file.field}) = 6
```

```
    then
```

```
        Date ("19" + {file.field}[5 to 6], {file.field}[1], {file.field}[3])
```

```
    else
```

```
        if
```

```
            {file.field}[3] = "/"
```

```
        then
```

```
            Date ("19" + {file.field}[6 to 7], {file.field}[1 to 2], {file.field}[4])
```

```
        else
```

```
            Date ("19" + {file.field}[6 to 7], {file.field}[1], {file.field}[3 to 4])
```

```
    else
```

```
        Date ("19" + {file.field}[7 to 8], {file.field}[1 to 2], {file.field}[4 to 5]);
```

---

The basic rule of database DLL compatibility: CRW is compatible with all PD\*.DLLs that ship with it or any time before it. But it is not (always) compatible with PD\*.DLLs that ship after it. This is because we tend to add new functionality to PD\*.DLLs as we go along (which may require new function calls to the DLL or new parameters to functions) and older builds of CRW do not expect these calls & parameters. (We take pains to make sure that CRW can handle all older versions of DLLs though - this is the more common case, e.g. a user installs PRO after VB but we still want to use his JET dlls.)

---

**E**

---

## F

---

"Error in Formula Code. Contact Crystal Services" message. The user is writing his own UFL. A bug in his UFL code.

---

"File cannot be recognized"

when opening an Access database within CRVB. Problem is solved by loading share (including share.exe /L:500" in the autoexec.bat). When installing, CR does prompt the user to see if he/she wants this included.

---

If you want to have different fonts depending on the outcome of an if-then-else statement you should have 2 formulas, one on top of the other. This is an example of the formulas:

Name: @Formula1

Text: if {file.field} = 2 then "This is OK" else ""

Name: @Formula2

Text: if {file.field} = 2 then "" else "This is OK"

The one formula would be in one font & the other in a second font

## G

---

A report will consistently GPF if you perform the following steps:

1. Create a subtotal
2. Create a Grand Total on same field
3. Remove the Grand Total field
4. Drag your Subtotal field into the Grand Total section
5. Attempt to use the Grand Total in a formula.

NOTE: If you do not attempt to use the grand total field in a formula all is well and the report will print correctly.

To fix your client's problem have him remove the grand total field then recreate the Grand Total freshly (ie: don't drag the subtotal field into the Grand Total).

This will be fixed in Version 3.

---

Windows Global Lock Error & file could not be recognized. This occurs when he is trying to open an XBASE database when opening a file with 450 fields. Is there a field limit?? He says that this works fine when he is opening a file with 30 fields. He has 3 meg of memory available. Probably is a limit that was raised in 2.0 slip.

---

**H**

---

**I**

---

**J**

---

**K**

---

## L

---

### Linking dBase to SQL

For any user trying to do this - linking between dBase (or any non-SQL) and SQL Server (or any SQL) tables - we really recommend linking in the opposite order to what this user is doing.

i.e. have the SQL tables look up the non-SQL tables, (define your links with SQL tables on the left of the dialog and dBase on the right).

This will avoid such limitations with Pioneer, and will also be much more efficient. What is happening with the way this user has constructed this report is that for every dBase record a separate SQL query needs to be built, sent to the server, executed, and lookup records returned. This is then all repeated for the next record.

We do some of this optimization for a report automatically, but what is probably happening here is:

- the user does not have a dBase index to look up on
- or, the user has made his linking so complex (by dBase -> SQL -> dBase, or (dBase -> SQL, dBase -> dBase) that we can't optimize

In the first case the user should generate more indexes, and in any case he should reexamine his linking. There are usually many different ways to link - and he/she may be able to simplify the report.

---

## M

---

Memo Fields: unable to print memo fields on multiple lines. Is the fix that concerns memo fields going to fix this as well?

If the report was made on a version of CRW when we did not support multiple lines for memo fields you should run the Verify database option. Its probably a safe bet to reinsert the field also at that point.

This is a problem with the Microsoft dll's which read xBase data. They always perform OEM to Ansi translation, so that the soft carriage return (CR) characters (0x8d in the OEM characters set) are translated into `\r` characters (0xec in the Ansi character set). We look for 0x8d characters, and treat them like CR's; however, in this case we don't see 0x8d, we see 0xec, and we leave it alone.

With (for example) Paradox, disabling translation of DOS memos preserves the 0x8d code, so we properly treat it as a CR. Our own xBase dll changes 0x8d to a normal CR before converting to Ansi, so it always works. But the Jet dlls, apparently, just translate from OEM to Ansi without checking for 0x8d.

Microsoft Access has the exact same problem with memo fields which contain soft CR's. I couldn't find an option there to cure the problem.

---

Internal error: **M**eta file not created

It looks like they must have run out of memory or disk space.

---

Multiple statements within an if-then-else formula:

```
If {file.field} = 20
then (first statement;
      second statement;
      third statement)
else 0;
```

---

If there is a problem with the "Xbase error" only when memo field inserted on report try doing a COPY TO within dBASE. Then try running the report again (changing the database file location to reflect the new database name). This usually takes care of the problem.

## N

---

### NULL DBPROCESS pointer Error

This is a confirmed bug in Pioneer's QLSS.DLL - it is using a DLL called W3DBLIB.DLL and so are other products such as Multilink. When we shut down (and therefore Pioneer shuts down) - QLSS.DLL does not detach cleanly from this DLL and affects other programs that are also using it.

We have contacted Pioneer and they do not have a fix for us yet. We are trying to get one from them.

What this means is that:

- they should use the ODBC SQL Server DLL that we also ship with CRW PRO - if they want to run simultaneously with programs such as Multilink
- if they are willing to run separately they can still use the QLSS.DLL
- this is why we have had no complaints from VB 3.0 users - the ODBC SQL Server DLL (which is the only one we ship in the bundle) doesn't have this problem - so please move users over to ODBC in this case

## O

---

### Oracle on George's machine

I have two directories containing different versions of config.sys and autoexec.bat, one directory called Backup2 is for oracle setup and the other (Backup) is a copy of a normal setup. If you want to run Oracle just copy the contents of c:\backup into c:\ (root directory) and reboot, then at the prompt type: Oracle, run windows in standard mode (win/s) and voila.

after exiting windows, at the D.O.S. prompt type: REMORA ALL

copy all contents from c:\backup to c:\ (root directory) and reboot, this will put you back where you started.

---

If you can't connect to ORACLE and are receiving a 3112 error you should run the SQL\*NET initializer. This is an oracle file netini.exe that comes in the Window's support files disk. This should allocate memory for remote sessions for the SQL\*NET driver. There is a parameter in the config.ora file that specifies how many connections to allow. (EG. remote\_connection = ).

Whenever you are connecting to ORACLE through Windows and receive a 3000-3999 error, you should run the net initializer. If you do not have the file contact ORACLE about obtaining it.

To test if SQLTCP or SQLSPX is really loaded, simply exit windows and type mem/c |more to view it loaded in memory. (or type SQLTCP or SQLSPX and if loaded, it will tell you that it already is)

---

## P

---

### Property - WindowParentHandle

We have concluded that the Report1.WindowParent call will not work, as you have indicated, and we suggest that you solve this problem in one of the following ways:

1) Use a Windows API call which will create an MDI child window within the parent window. Basically, you would create an MDI frame and then you would place our Crystal Print Window inside of this frame but you would eliminate the control menu, the max and min buttons, the borders etc. etc. Visual Basic has problems using MDI Child windows and it seems that this is the only workaround. For a running example of this please dial into our BBS and download an application called VBCHILD.ZIP. Our BBS number is (604)681-9516.

2) If you want our print window to be a child of the parent window, as you have tried to do with the Report1.WindowParentHandle = Form1.hWnd, then you will need to make yet another Windows API call which will "refresh" the Crystal Print Window immediately after printing to screen. This call will bring the focus back to our Crystal Print window as well as eliminate any transparency problems.

---

### PESelectPrinter:

If you pass 0 for any or all of the driver name, printer name or port name parameters, we set the printer in the report to the Windows default printer (ignoring the devmode parameter).

If you give strings for all three, we try to point to the specified printer; if it's not installed exactly as specified by these arguments, the call will fail. For example, if you have the WhizBang 5000 installed on LPT1:, but ask for the WhizBang 5000 on LPT2:, the call will result in PE\_ERR\_INVALIDPRINTER.

In the first case, in all released versions, the devmode parameter is ignored. This means you can't produce the state you get in CRW by going into the printer selection dialog, choosing Default Printer, and changing other settings. PESelectPrinter only allows you to have a pure default printer, or a completely specified one. In CRW 3.0, the devmode parameter is used, and you can produce the same situation as in CRW.

In the second case (a specific printer), you can also pass a non-zero DEVMODE, which will override the DEVMODE settings (e.g. paper size, orientation) in the Control Panel. Note: it's a relatively advanced bit of Windows programming to set up a DEVMODE, so normally people will pass 0 for this parameter.

---

### Printer Save Settings

For your information Crystal Reports has 3 save settings regarding printers.

The first, is "True Use default printer." This is set when you create a report and then just save it. It will use whatever settings are in the Control Panel.

The second, is "Use default printer but with these settings." This comes about when you create a report and then go into Select Printer, set up various options and then save the report.

The third, is "Use a specific printer."

You can change the 2nd setting to the 3rd and vice versa but once you leave the first setting the only thing you can do is recreate the report.

Q

---

## **R**

---

### Report1.DataFiles

You cannot use this property for tables within \*.MDB files. It seems that you can only set the location of the datafile itself and there is no way to actually point at a specific table.

**S**

---

**T**

---

## U

---

### User Defined Size

Some printers will allow the user to select **User Defined Size** when in Print Setup.

The page size that will appear on the CRW screen is dependent on the settings in Control Panel - Printers.

Select your particular printer and Setup.

For Paper Size: specify User Defined Size

A dialog box will appear where you can specify the size and units you require. **Note:6** the units are 0.1mm or 0.01 inch. These are small increments; 8.5 inch paper width would have 850 in the box.

When you return to CRW, run Select Printer again to see the updated size.

---

### UFL's

"integer" is not allowed as a

parameter type to user defined functions. The only numerical typed allowed is "number". So, the function parser actually kicked the functions out, which is why they weren't recognized when they were used.

## V

---

Bug in the VBX which means that if you set the SelectionFormula (or GroupSelectionFormula) property, and the report already has a selection formula of the same kind, the one in the property overrides that in the report, rather than being appended to it as it should be.

I've fixed the bug. Greg will decide what to do with the fix. - rick

---

There is a bug in crystal.vbx related to setting an array property, such as Formulas, DataFiles, etc. Sometimes, when more than one element in the array is set, our vbx will corrupt a VB data segment, which can lead to a GPF in VB.

I have found and fixed the problem. New vbx on BBS.

---

The VBX is loaded when the form which holds it is loaded - so, if this is not the main form, it won't be loaded when the app starts.

CRPE is loaded the first time you execute Action = 1, in the case of CR for VB. In CRW 2.0, CRPE is loaded at the same time as the VBX.

VB is a real memory hog. We've found that, on a 4 MB machine, there's so little memory left for CRPE that it thrashes (technical term) while printing.

The only solution is to get more memory!

It's possible that performance is better if the VB app is compiled, rather than being run inside VB - but I wouldn't count on it!

---

When you click on Database, Verify Database and there are major changes made (database very different from when report created) it will ask you if you want to fix up the report.

If there are minor changes made you will receive a message that the "Database is NOW up to date".

If there are no changes at all made you will receive the message "Database is up to date".

---

The user is receiving a VB stack fault error. The stack size in VB is 20K and cannot be changed. Usually when using the Crystal VBX this is adequate, but in a few cases like Oracle (whose DLLs seem to require a lot of stack space) it may not be if the user has a large app doing a lot of other things at the same time. The user may need to experiment with simplifying his/her application until this error goes away, and then rebuild the application so that as little as possible (few dialogs etc.) is going on at the same time as he/she starts printing.

You can also refer the user to his VB Programmer's Guide, pages 250-1, "Minimizing Stack Use".

Warning: This user may not be running out of stack - it's hard to know by the error message. You may need to discuss further with him, and try some workarounds.

---

VBX fixes

Here are the new vbx files for CRWPRO and the MicroSoft Build.

Please zip up each accordingly and place up on the BBS. See below for a list of fixes.

Note: The attachment- The MS build has been called crystalm.vbx but should be crystal.vbx in your final zip file.

The problems fixed are:

- 1) bug with array properties
- 2) bug with SelectionFormula and GroupSelectionFormula properties, where the piece given through the vbx replaced the old formula, rather than being added to it.

W

---

**X**

---

**Y**

---

**Z**

---