

VB3 – VB6 Conversion

Converting Projects From VB3 – VB6

Copyright© 1999-2001 Denver Tax Software, Inc. All Rights Reserved.

DECLARATIONS

Windows API

DtsWin.dll, if possible, otherwise use Dan Appleman's CD. It is best to "wrap" Windows API calls to keep the changes in one place in a DLL file rather than many places throughout the project(s).

DLL Calls

For calls to Crescent Quick Pack, refer to online help to see what the updated declaration should be.

Public Declarations In Frm Files

Public declarations should only be in BAS files. If a public declaration is in a FRM file, it will probably cause an error, but that error might be very difficult to find.

Boolean

In VB3 Integer types were used as boolean types. With VB6 there is an actual Boolean type. Recommend changing As Integer to As Boolean. Don't do a global change. An integer of 3 could be true or 3. A boolean variable could be "true" but never "3"!

Controls

Overview

Let VB6 attempt to convert. Check log files. Many controls will convert.

Edit VB3 Files First

For some controls it is helpful to convert to the new control name before VB6 attempts the conversion.

Example –

VB3 code-

```
Begin Grid grdDisplay
  Cols      = 4
  FontBold  = -1 'True
```

VB6 replacement-

```
Begin vsFlexLib.vsFlexArray vsgrdDisplay
```

Height = 4560
Left = 720

Procedure:

1. Create a picture placeholder control on the VB3 form, e.g. PicGrdDisplay.
2. Print out the properties of the control to be replaced.
3. Within the VB3 IDE, replace all references of the old control with the placeholder control name. E.g, replace grdDisplay with PicGrdDisplay.
4. Delete the old control from the form.
5. Repeat steps 1 – 4 for all controls that need to be replaced.
6. Open the .mak file within the VB6 IDE.
7. Add any needed components to replace the old controls.
8. Put the new control on the form. For this example, lets call it vsGrdDisplay.
9. Within the VB6 IDE, replace references of the placeholder picture control with the name of the new control. In this example, replace PicGrdDisplay with vsGrdDisplay.
10. Delete to picture placeholder control.
11. Set the initial properties of the new control from the printout done in step 2.
12. Repeat steps 7 – 11 for all controls to be used as replacements.

Funny Looking Controls

Quick Pack:

Some of the Quick Pack controls looked strange. The solution is simple for some of these controls. Change the control's size, and return it to the original size.

For the Crescent Caption controls, it is test to reset the top, left, right and bottom margins to 4. The dimensions should be re-entered by typing the dimisions into the property sheet rather than using the mouse to resize the controls.

Replace the option control with its 32 bit version.

The meter control – After changing PercentValue, add DoEvents to display the change. You may want to replace the control if problems occur.

verStamp.exeProductVersion

The above commonly used property does not exist. Replace it with app.major & app.minor, etc.

Toolbar

Redo the toolbar using the VB6 features. Need MS Win Com Ctrl 5.0(SP2) & MS Win Com Ctrl 2 5.0(SP2).

vsGrid

To justify properly use (vs)CellAlignment=

chkReport_Click

The VB3 events that determine what reports are to print appears to put the system in an infinite loop. Refer to WLB to see how that was fixed.

!

Change the '!' to a ".".

OutOfRange

OutOfRange was a custom event that was in some Crescent controls for an old version that is not available for the most current version. Consider modifying program, and using Validate event, possibly with CausesValidation property.

Consider using RangeWarning from SSP. Review how it is implemented.

Out Of Stack Space – Crescent Option/Radio Buttons

In VB6 having code in the Click() event can cause an infinite loop resulting in Out Of Stack Space errors. The solution is to move the code to the control's LostFocus() event.

ILLEGAL OPERATION VB6.EXE 017f:005810d8

Remove the version stamper control from the form using VB3. After that the error, will not appear.

Sheridan Datawidgets Error 381, Invalid Property Array Index.

For Datawidgets 1, the Row property was the absolute row in the list. For Datawidgets 3, Row is the visible row in the list. The best way to move through the list is to use the MoveNext method.

Functions & Procedures

CALL ...

Some calls to procedures may have to be made with CALL <procname>. This was true with one of the Quick Pack procedures.

QROUND

If Quick Pack's QROUND is used, replace it with VB's ROUND().

LEN()

To determine the length of some user defined types, LENB() may be more accurate than LEN().

ISORTT

I have experienced problems with Crescent's ISORTT and BCopyT. I have used SORTT instead with some re-programming. BcopyT can be easily replaced w/ VB6 code.

Setup

Have Wise check to see what dlls are used. Close out TSRs, including virus checker.

DTS Specific Programming

Old Versions

When reading old version files, change version numbers from xx to 19xx. E.g., WLB98.10 changes to WLB1998.10. See function in WLB.

Retrieving File Data Created in VB3

Overview

We have run into several problems. In VB3 we typically would save data that was in user defined types (UDT) to a file using Crescent's PowerPak's binary file routines. Often this information could not be easily retrieved in VB6. See ConvertVB3Data2VB6.vbp in our test folder.

Dates

In VB3 we stored dates as variants. In VB3 the date information is in the first 8 bytes of the date – variant. In VB3 a date variant takes either 10 or 16 bytes. The first two bytes are for the Variant code. The next eight bytes of the VB3 variant translate into the VB6 Date data type. Any other bytes left in the VB3 variant should be considered as a placeholder that is not used.

VB3 –

```
Type tagBirthday
    dtBDay as Variant          '16 bytes
    cBDayGift Cost as Currency '8 bytes
    sDesc as string * 3
```

End Type

VB6

```
Type tagBirthday
    bVariantDateCode(0 to 1) as byte    '2 bytes Don't use this info.
    dtDate as Date                      '8 bytes This is the date
    bFillerPartOfVariant(0 to 5) as byte '6 bytes, This might not be needed
    cBDayGift Cost as Currency          '8 bytes
    sbDesc(0 to 2) as byte
```

End Type

It is a good idea to convert dates as Variant to Date type. Change Cvdate() to Cdate().

Crescent File Access Routines

Change to the VB6 file access routines. I have had some problems w/ the Crescent routines, but not the VB6 routines, Open, Get, Put, Close.

Unicode Strings

VB3 represents strings with 1 byte per character. VB6 may represent strings with 2 byte per character to impliment Unicode. This presents a problem in converting data created in VB3 that contained strings to VB6 applications. For more info on this, see "Unicode and the byte data type" article on the MSDN CD. The VB6 UDT that receives the information from the VB3 created file must use a byte array that is the same length as the VB3 string.

Use StrConv(<byte string>, vbUnicode) to convert a byte string from a VB3 file to a VB6 Unicode string.

Refer to the Social Security Planner's retrieve file routines.

It is also a good idea to "brand" files created with VB6 to determine whether conversion is needed.

Installation Considerations

Wise Included VB6 Files That Are Sometime Out Of Date

Wise keeps common VB6 runtime files in F:\Wise\RUNTIME\VB6. These do not get updated with VB Service Packs. After installing a VB Service Pack or update, files in F:\Wise\RUNTIME\VB6 should be updated as well. This could also apply to other sub directories of F:\Wise\RUNTIME.

Registering Components

Upon installation some installed components might not be registered. Make sure Regsvr32.exe is also included so the user can manually register, if needed.

WebDeploy

In the WebDeploy configuration screen, "File Transfer Via:" should be FTP, not HTTP.

On WSC, the userid webdeploy must get the key for the library where the WebDeploy files reside.

Database Considerations

Workspace

Workspace object is needed with VB6.

Obsolete Or Changed Database Commands(See MSDN Obsolete Features in DAO)

OpenDatabase statement [DBEngine.OpenDatabase](#) method

Database.OpenQuerydef method [Querydefs](#) collection

Database.DeleteQuerydef method [Delete](#) method

CreateDatabase statement [DBEngine.CreateDatabase](#) method. One argument will specify what version of the DAO is to be used to save the database.

Database.Opentable method [OpenRecordset](#) method

To reference a record set table item use:

rs(FIELD_NAME_VARIABLE) or rs![Client Code]. FIELD_NAME_VARIABLE="[Client Code]"

DAO

In the VB IDE Projects – Reference. Choose Microsoft DAO 2.5/3.5 Compatibility Library. For Access 97 databases choose Microsoft DAO 3.5 Object Library if backward compatibility is not an issue. Microsoft DAO 3.6 Object Library is for Access 2000.

Error 3061

If one gets error 3061 when opening a recordset, use dbOpenDynaset even if the recordset is a snapshot. E.g.,:

```
Set rsCount = dbKey.OpenRecordset(sCountSQL, _  
    dbOpenDynaset)
```

Error 3315

Field <> can't be a zero length string. Field property AllowZeroLength is set to false. Either set it to true or make sure that what goes into that field is not zero length.

Refresh Method Does Not Refresh Data Control

Change Connect property of data control to that of the DAO Access Library. Thus, for Microsoft DAO 3.5 Object Library (Access 97) the Connect property should be Access, not Access 2000.

Deactivate Bug

VB6 hangs and gets a VB6.EXE Illegal Operations error in 017f:00417ca when unloading a form from <form>_deactivate. The purpose of using deactivate is to know when another form has focus.

The solution is to put the unload command for one form before the show method of the next.

```
Unload dlgBasisReportOpt  
dlgMethod.Show vbModal
```

Bookmarks - Grids

For VB6 the bookmark has replaced the row property in grid controls. The Row property refers to the visible row not the absolute row. E.g., if there are 20 rows (0 – 19) and the first 15 (0 – 14) are visible, row 15 is an Invalid Property Value, error 381.

To use bookmarks to replace the VB3 Row property.

1. Define the 0 row (absolute) book mark: DIM bmRow0 as Variant
2. Initialize bmRow0:
Grid.MoveFirst
bmRow0=Grid.Bookmark

3. Naviage the Grid:
 For j = 0 to Grid.Rows-1
 Grid.Bookmark=bmRow0 'Return to reference point
 Grid.Bookmark=Grid.Getbookmark(j)
 Next j
4. Return pointer to first row: Grid.Bookmark=bmRow0

To keep the screen from flickering, one might want Grid.Redraw set to false before step 1 and set to true after step 4.

Datawidgets Rows Property

With VB3 this could be set at runtime. Probably can be eliminated. When trying to set .Rows at runtime gets an error.

Recommendation: Search .Rows, and comment out lines.

Datawidgets RowLoaded Event – Needs to be removed.

DATAGRID -

This was used in VB3 when the unbound datagrid was used as a list box. Make sure the grid's DataMode property is set to AddItem. Replace RowLoaded event with code like:

```
For iRemovedBlockIdx = 0 To UBound(gaRemBlocks)
    dlgRemBlocks.SSdgRemBlocks.AddItem gaRemBlocks(iRemovedBlockIdx)
Next
```

COMBO BOX (Unbound) –

Use the controls property page to add a column.

Before showing the form include something like this:

```
With dlgSellShares.SScboSellMethod
    .Columns(0).Style = ssStyleComboBox
    .AddItem SELL_LIST_SPEC_ID, 0
    .AddItem SELL_LIST_FIFO, 1
    .AddItem SELL_LIST_AVG_SNG, 2
    .AddItem SELL_LIST_AVG_DBL, 3
End With
```

Comparing BookMarks

If bmOne = bmTwo Then -> will generate an error.

Use: If Cstr(bmOne) = Cstr(bmTwo) Then -> will work.

Sheridan Grid – Fill In Unbound Grid

The FieldSeparator property should be set to tab.

```
For l = 0 To 32
    SSDBG1.AddItem "Hello" & Chr$(9) & "World"
```

Next I

Chr\$(9), vbTab, indicates a column separator

Crystal

1. Open and Save each report with the newest version of Crystal.
2. Refer to Sub RunReport in BK. This is a candidate for an ActiveX DLL.