PREFACE

I am placing this code and documentation in the public domain in the hopes that others may find it useful. You are free to use, modify and distribute it as you see fit. This code is provided on an as-is basis; I have tested much of it but it is not guaranteed to be bug-free. If you find errors or have suggestions for improvement, you can send them to me if you'd like.

Sharon F. Dooley
January 2, 1992
CompuServe ID 70740,2330

PAL and PARADOX are trademarks of Borland. Visual Basic is a trademark of Microsoft.

A VISUAL BASIC -- PARADOX ENGINE INTERFACE

Introduction

This Visual Basic interface to the Paradox Engine insulates the VB programmer from some of the quirks of the Paradox Engine as well as from some of the differences between VB and a C DLL. It also permits the VB programmer to work with VB's native data types while simultaneously allowing the Paradox user or PAL (Paradox Application Language) programmer to work with the native Paradox data types. For example, VB has a wealth of date manipulation functions that want the VB date-serial date format. The PAL date functions naturally expect a Paradox date format. The interface layer translates to and from the database format as data is retrieved from or stored in a Paradox table. The separate interface layer also provides a measure of independence from the particular DBMS.

VB programmers can access the engine by calling a set of functions which parallel the engine's C functions. For convenience, these routines have the same name as the engine functions with the prefix VB. For example, VBPXGetShort calls the engine's PXGetShort.

This document assumes that the reader has a working knowledge of the engine functions, Paradox database constructs, and Visual Basic. It also assumes that user of the package will be moderately curious and willing to look at the supplied code if the documentation isn't clear.

This package contains two files in addition to this documentation. The first, VBPXGLOB.TXT, contains declarations of global constants that the VB programmer can use. These global constants include TypeDefs for the various objects the engine uses: handles, buffers, etc., mnemonics for the engine's error codes, and some of the 'words' used by engine functions such as SEARCHFIRST, SEARCHNEXT. The interface layer assumes these definitions are global. If you do not want to use them, add them to the module that contains VBPXENG.TXT. If you also want to use them, add them to your global module. The second file is VBPXENG.TXT. This file contains a few constants used only by the VBPX... routines, the declarations for the Paradox Engine DLL, and the VBPX... functions. Note that this is a TEXT file. To use it in your VB application, you should create a new module file. Next, use the load text option from the code menu to place this text into the file. Like their Engine counterparts, the VBPX... routines all return an integer containing a Paradox status code. Your programs should always test this and handle the error if there is one. I have included a function that puts up a message box with the error message text. You will probably find this useful for debugging.

The following sections list the VBPX... functions together with their arguments. For convenience, I have organized them into the categories used in the Paradox Engine documentation. If the function simply calls a Paradox function, only the name of that function is given. In this case, the VB function simply passes the arguments along to the engine. You can read the description of the function in the Engine documentation. If the function transforms data or is different in some way from the Engine function that it calls, there is a brief description of the differences.

INITIALIZATION AND FINALIZATION FUNCTIONS

Function VBPXExit() As Integer

Invokes PXExit.

Function VBPXGetDefaults (swapSize As Integer, maxTables As Integer, maxRecBufs As Integer, maxLocks As Integer, maxFiles As Integer, sortOrder As String) As Integer

Invokes PXGetDefaults

Function VBPXNetInit (netNamePath As String, netType As Integer, UserName As String) As Integer

Invokes PXNetInit; not generally used in Windows

Function VBPXSave () As Integer

Invokes PXSave

Function VBPXSetDefaults (bufSize As Integer, maxTables As Integer, maxRecBufs As Integer, maxLocks As Integer, maxFiles As Integer, sortOrder As String) As Integer

Invokes PXSetDefaults; note that there are defined constants for sort order in VBPXGLOB.TXT. This may not be useful because you need to run the engine configuration program to tell Windows what your maxima are, but the engine documentation is unclear, so here it is.

Function VBPXWinInit (ClientName As String, ShareMode As Integer) As Integer

Invokes PXWinInit

TABLE FUNCTIONS

Function VBPXTblAdd (srcTableName As String, destTableName As String) As Integer

Invokes PXTblAdd

Function VBPXTblClose (table As TABLEHANDLE) As Integer

Invokes PXTblClose

Function VBPXTblCopy (srcTableName As String, destTableName As String) As Integer

Invokes PXTblCopy

Function VBPXTblCreate (TableName As String, NumFields As Integer, Fields() As String, Types() As String) As Integer

Fields and Types are simple VB String arrays of NumFields entries. This routine handles all the conversion of these arrays into the Engine's required array of pointers to character. Invokes PXTblCreate.

Function VBPXTblDelete (TableName As String) As Integer

Invokes PXTblDelete

Function VBPXTblEmpty (TableName As String) As Integer

Invokes PXTblEmpty

Function VBPXTblMaxSize (maxTblSize As Integer) As Integer

Invokes PXTblMaxSize

Function VBPXTblName (table As TABLEHANDLE, TableName As String) As Integer

Returns the table name as a VB String

Function VBPXTblOpen (TableName As String, table As TABLEHANDLE, indexId As Integer, saveEveryChange As Integer) As Integer

Invokes PXTblOpen. Remember that Paradox uses 1 for TRUE, so use a constant 1 for saveEveryChange if you want it, otherwise use a zero.

Function VBPXTblRename (srcTableName As String, destTableName As String) As Integer
Invokes PXTblRename

RECORD FUNCTIONS

Function VBPXRecAppend (table As TABLEHANDLE, Record As RECORDHANDLE) As Integer
Invokes PXRecAppend
Function VBPXRecBufClose (Record As RECORDHANDLE) As Integer
Invokes PxRecBufClose
Function VBPXRecBufCopy (SrcRecord As RECORDHANDLE, DestRecord As RECORDHANDLE) As Integer
Invokes PXRecBufCopy
Function VBPXRecBufEmpty (Record As RECORDHANDLE) As Integer
Invokes PXRecBufEmpty
Function VBPXRecBufOpen (table As TABLEHANDLE, Record As RECORDHANDLE) As Integer
Invokes PXRecBufOpen
Function VBPXRecDelete (table As TABLEHANDLE) As Integer
Invokes PXRecDelete
Function VBPXRecFirst (table As TABLEHANDLE) As Integer
Invokes PXRecFirst
Function VBPXRecGet (table As TABLEHANDLE, Record As RECORDHANDLE) As Integer
Invokes PXRecGet
Function VBPXRecGoto (table As TABLEHANDLE, Record As RECORDHANDLE) As Integer
Invokes PXRecGoTo
Function VBPXRecInsert (table As TABLEHANDLE, Record As RECORDHANDLE) As Integer
Invokes PXRecInsert
Function VBPXRecLast (table As TABLEHANDLE) As Integer
Invokes PXRecLast
Function VBPXRecNext (table As TABLEHANDLE) As Integer
Invokes PXRecNext
Function VBPXRecNum (table As TABLEHANDLE, RNum As RECORDNUMBER) As Integer
Invokes PXRecNum
Function VBPXRecPrev (table As TABLEHANDLE) As Integer
Invokes PXRecPrev
Function VBPXRecUpdate (table As TABLEHANDLE, Record As RECORDHANDLE) As Integer
Invokes PXRecUpdate

FIELD FUNCTIONS

Function VBPXFldBlank (Record As RECORDHANDLE, Field As FIELDHANDLE) As Integer
Unlike the engine function, which places the truth value of the blank comparison in a separate variable, this function returns TRUE (-1) if field is blank, 0 if field is not blank, error code otherwise. It invokes PXFldBlank.
Function VBPXFldHandle (table As TABLEHANDLE, fldName As String, Field As FIELDHANDLE) As Integer
Invokes PXFldHandle

Function VBPXFldName (table As TABLEHANDLE, Field As FIELDHANDLE, fldName As String) As Integer

Invokes PXFldName

Function VBPXFldType (table As TABLEHANDLE, Field As FIELDHANDLE, fldType As String) As Integer

Calls PXFldType and places a VB String in fldType

Function VBPXGetAlpha (Record As RECORDHANDLE, Field As FIELDHANDLE, dest As String) As Integer

If the PXGetAlpha call is successful, strips the null terminator and returns a true VB string in dest.

Function VBPXGetCurrency (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Currency) As Integer

Converts a Paradox Double to a VB Currency

Function VBPXGetDate (Record As RECORDHANDLE, Field As FIELDHANDLE, dateval As Double)

Gets a Paradox date and converts it to a VB Date Serial

Function VBPXGetDoub (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Double) As Integer

Invokes PXGetDoub

Function VBPXGetLong (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Long) As Integer

Invokes PXGetLong

Function VBPXGetShort (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Integer) As Integer

Invokes PXGetShort

Function VBPXPutAlpha (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As String) As Integer

Invokes PXPutAlpha

Function VBPXPutBlank (Record As RECORDHANDLE, Field As FIELDHANDLE) As Integer

Invokes PXPutBlank

Function VBPXPutCurrency (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Currency) As Integer

Converts a VB Currency variable to a double and puts it into the record buffer with PXPutDoub

Function VBPXPutDate (Record As RECORDHANDLE, Field As FIELDHANDLE, dateval As Double) As Integer

converts a VB date serial to a Paradox Date field and puts it into the record buffer with PXPutDate

Function VBPXPutDoub (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Double) As Integer

Invokes PXPutDoub

Function VBPXPutLong (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Long) As Integer

Invokes PXPutLong

Function VBPXPutShort (Record As RECORDHANDLE, Field As FIELDHANDLE, Value As Integer) As Integer

Invokes PXPutShort

KEY AND INDEX FUNCTIONS

Function VBPXKeyAdd (tblName As String, nflds As Integer, Fields() As FIELDHANDLE, IndexType As Integer) As Integer

Invokes PXKeyAdd. Constants for IndexType are defined in VBPXGLOB.

Function VBPXKeyDrop (tblName As String, indexId As Integer) As Integer

Invokes PXKeyDrop

SEARCH FUNCTIONS

Function VBPXSrchFld (table As TABLEHANDLE, Record As RECORDHANDLE, Field As FIELDHANDLE, SearchMode As Integer) As Integer

Invokes PXSrchFld. Constants for SearchMode are defined in VBPXGLOB.

Function VBPXSrchKey (table As TABLEHANDLE, Record As RECORDHANDLE, nflds As Integer, SearchMode As Integer) As Integer

Invokes PXSrchKey. Constants for SearchMode are defined in VBPXGLOB.

PASSWORD/SECURITY FUNCTIONS

Function VBPXPswAdd (Password As String) As Integer

Invokes PXPswAdd.

Function VBPXPswDel (Password As String) As Integer

Invokes PXPswDel.

Function VBPXTblDecrypt (TableName As String) As Integer

Invokes PXTblDecrypt.

Function VBPXTblEncrypt (TableName As String, Password As String) As Integer

Invokes PXTblEncrypt

Function VBPXTblProtected (TableName As String) As Integer

Invokes PXTblProtected and returns TRUE if table is protected, FALSE if not, otherwise, it returns an error code

INFORMATION FUNCTIONS

Function VBPXKeyNFlds (table As TABLEHANDLE, nKeyFlds As Integer) As Integer

Invokes PXKeyNFlds

Function VBPXRecNFlds (table As TABLEHANDLE, nflds As Integer) As Integer

Invokes PXRecNFlds

Function VBPXTblExist (TableName As String) As Integer

Invokes PXTblExist and returns TRUE if table exists, FALSE if table doesn't exist, otherwise, a Paradox Error Code

Function VBPXTblNRecs (table As TABLEHANDLE, NRecs As RECORDNUMBER) As Integer

Invokes PXTblNRecs

NETWORK/CONCURRENCY FUNCTIONS

Function VBPXNetFileLock (fileName As String, lockType As Integer) As Integer
    Invokes PXNetFileLock
Function VBPXNetFileUnlock (fileName As String, lockType As Integer) As Integer
    Invokes PXNetFileLock
Function VBPXNetRecGotoLock (table As TABLEHANDLE, PXlock As LOCKHANDLE) As Integer
    Invokes PXNetRecGotoLock
Function VBPXNetRecLock (table As TABLEHANDLE, PXlock As LOCKHANDLE) As Integer
    Invokes PXNetRecLock
Function VBPXNetRecUnlock (table As TABLEHANDLE, PXlock As LOCKHANDLE) As Integer
    Invokes PXNetRecUnlock
Function VBPXNetRecLocked (table As TABLEHANDLE) As Integer
    Invokes PXNetRecLocked and returns TRUE (-1) if the current record of table is locked, FALSE if it is not locked, and otherwise an error code.
Function VBPXNetTblChanged (table As TABLEHANDLE) As Integer
    Invokes PXNetTblChanged, and returns TRUE (-1) if table has changed. Returns FALSE if table hasn't changed; otherwise it returns an error code.
Function VBPXNetTblLock (table As TABLEHANDLE, lockType As Integer) As Integer
    Invokes PXNetTblLock
Function VBPXNetTblRefresh (table As TABLEHANDLE) As Integer
    Invokes PXNetTblRefresh
Function VBPXNetTblUnlock (table As TABLEHANDLE, lockType As Integer) As Integer
    Invokes PXNetTblUnlock
Function VBPXNetUserName (UserName As String) As Integer
    Invokes PXNetUserName and, if successful, returns the name as a VB string. Otherwise, it returns an error code.

ERROR FUNCTIONS

Function VBPXERRMSG (errcode As Integer) As String
    Calls PXErrMsg and returns the text for a Paradox Error code as a VB string
Function VBPXNetErrUser (UserName As String) As Integer
    Calls PXNetErrUser and returns the user name as a VB string.
Function showPDOXError (errcode As Integer) As Integer
    puts up a message box with the stop icon and the text of the error. Program stops after you click ok on the msg box so you can explore with the debugger, then, if you continue, it calls VBPXExit so you can get out cleanly.

BLANK FIELD FUNCTIONS

Sub BLANKSHORT (X As Integer)
    Places a Paradox Short Blank in X
Sub BLANKDATE (X As Double)
    Places a Paradox Blank Date in X
Sub BLANKLONG (X As Long)
    Places a Paradox Blank Long in X

Note: VBISBLANK... Functions return a VB TRUE if the the field is blank otherwise they return a VB FALSE

```
Function VBISBLANKSHORT (X As Integer) As Integer
Function VBISBLANKLONG (X As Long) As Integer
Function VBISBLANKDOUBLE (X As Double) As Integer
Function VBISBLANKDATE (X As Double) As Integer
```

See also VBPXFldBlank, VBPXPutBlank