CHROME COMPUTING
Bob Kubelka
CallBtrv() - version 1.0
05/01/92

This is a sample program which demonstrates accessing Btrieve files using a VB function called CallBtrv(OpCode,FileNo,KeyNo). This function was written by Bob Kubelka of Chrome Computing, Vista, CA 92083 (619) 941-2893. You may use this function and distribute it freely. The program requires Visual Basic, VBRUN100.DLL and WBTRCALL.DLL (from Novell).

This function is designed to do the grunt work of setting-up Btrieve data structures, such as the position block, data buffer and all other parameters needed to call WBTRCALL.DLL. Consequently, it is extremely easy for you to to use Btrieve without the need of setting-up all these parameters and making a call directly to Btrieve every time your program needs to call Btrieve. Instead all you need to do is put your data in a buffer (if doing an insert or update operation) or put the key you wish to find in KeyVal (if doing a get operation) and call CallBtrv passing just three parameters: the operation code, the file number and the key number.

The source code is heavily remarked. You may want to remove the remarks once you understand it - this makes smaller EXEs.

To run this program correctly, first edit the form1.load event and change the 'Path' to the directory where you have installed the project. Run the program. Click 'Init' to establish communication with Btrieve. Then click 'Open' to open the sample file. Then you can get, update, insert and delete records.

To get a record using a key, enter the key in the 'Keyval' edit box. Also specify which key you want to use. The Sample file uses three keys (0,1 & 2). The returned data is shown in edit controls. To update a record, simply get a current record, make the changes in the edit boxes and click 'Update'.
To insert a new record, enter the data and click 'Insert'.
To delete a record, first GET the record you want to delete and click 'Delete'.

Note that the sample file has keys that are strings. To have keys which contain integers or other data types, create a user defined type such as:

```
Type USER_TYPE
  Customer Number as integer
  Customer Name as string * 30
end type
Global UserType = USER_TYPE
```

Create another user defined type which has just one element which is a string defined as having a length at least as long as USER_TYPE.

```
Type USER_BUFFER
  Buffer as string * 32
End Type
Global UserBuffer = USER_BUFFER
```

Before calling CallBtrv(), issue the following code:

```
LSet UserBuffer = UserType
KeyVal = UserBuffer.Buffer
CallBtrv(n,n,n)
```

The Key value is now in KeyVal, which is a string. You can now call GET functions. To get the key value returned by Btrieve, do the opposite:

```
UserBuffer.Buffer = KeyVal
LSet UserType = UserBuffer
```

Another way of doing this is to change KeyVal, which is the string used by CallBtrv() for the Btrieve key value, to a user defined type such as:

```
Type KEY_BUFFER
  Key as string * 128
End Type
Global KeyBuffer = KEY_BUFFER
```

Make the lenth 128 which Btrieve wants. Then you can insert code such as:

```
LSet KeyBuffer.Key = UserBuffer
CallBtrv(n,n,n)
```

The sample program does not demonstrate record locking. This is simple to implement and it is left as an exercise for you.

The operation codes for extended record operations are left as an excercise for you to implement in CallBtrv().

These are just some ways you may want to modify the code to suit your needs. Let me know if this helps you. Also, if you make improvements, I'd like to know.