

(text continued from page 42)

would be nice to get rid of the CtlInfo table. This turns out to be entirely possible, but not without a couple of snags.

resize2.h (Listing 6) declares makeResizable2(). Instead of a Resize pointer, its second parameter is simply the ID of the pivot control.

The implementation of makeResizable2() is in resize.c (Listing 5). This function dynamically allocates and initializes a Resizer structure and then calls makeResizable(). In other words, it uses the exact same resizing machinery that I've described, the only difference being the automatic calculation of the CtlInfo table.

**Listing 5: resize.c — continued**

```

resizeWndProc2, hwnd, msg, wParam, lParam );
}

BOOL makeResizable2( HWND hwnd, int idPivot ) {
    HWND hwndChild;
    RECT rcPivot;
    Resizer *pResizer;
    int iChild = 0;
    HWND hwndPivot = GetDlgItem( hwnd, idPivot );

    if ( !IsWindow( hwndPivot ) ) {
        return FALSE; //*** FUNCTION EXIT POINT
    }
    pResizer = calloc( 1, sizeof( Resizer ) );
    if ( 0 == pResizer ) {
        return FALSE; //*** FUNCTION EXIT POINT
    }


    for ( hwndChild = GetTopWindow( hwnd ); IsWindow( hwndChild );
        hwndChild = GetNextSibling( hwndChild ) )
    {
        pResizer->nNumCtrls++;
    }

    pResizer->pCtlInfo = calloc(
        pResizer->nNumCtrls, sizeof( CtlInfo ) );
    if ( 0 == pResizer->pCtlInfo ) {
        free( pResizer );
        return FALSE; //*** FUNCTION EXIT POINT
    }
    GetWindowRect( hwndPivot, &rcPivot );

    hwndChild = GetTopWindow( hwnd );
    for ( iChild = 0; iChild < pResizer->nNumCtrls; ++iChild ) {
        CtlInfo *pCtlInfo = &pResizer->pCtlInfo[ iChild ];
        RECT rcChild;
        GetWindowRect( hwndChild, &rcChild );
        pCtlInfo->nCtlId = GetWindowID( hwndChild );
        pCtlInfo->nHorzAlignment = none;
        if ( rcPivot.right <= rcChild.left ) {

```

RS232-TOOLS



www.adontec.com

### SUPERCOM

Windows 2000/NT/95/98/3.x, DOS or OS/2

*for MS C/C++, Visual C++, VisualBasic, Borland C/C++, C++ Builder, Borland Pascal, Delphi, IBM C/C++, WATCOM C/C++*

SuperCom is the development tool for event driven communication software. This means high data security and highest transmission speed. Developers use one SuperCom API for RS-232, ISDN(CAPI) and TCP/IP among different languages and operating systems. SuperCom supports popular protocols like ZMODEM, X/MODEM, ASCII, Hayes comp. Modem, RS-485, Multiserial boards...

Looking for ZMODEM over ISDN or ZMODEM over IP? Check our web site for detail info.

**OPTIONAL AVAILABLE INDUSTRIAL PROTOCOLS**

3964/R Protocol Engine ActiveX       LSV/2

**SUPERCOM ACTIVEX FOR WIN 2000/NT/95/98**

Dialing up using Hayes compatible modems and file transfers using popular protocols like: ZMODEM, Y/MODEM and ASCII. Events: OnComm, OnModem and OnFileInfo. It can be used by any NT/95/98 compiler (e.g. Visual Basic, Delphi, C++ Builder, Visual C++, MS Office).



**SUPERMONITOR**

for DOS, Windows 3.x/95/98

The flexible and versatile solution for your serial data and protocol analysis. You are now able to considerably reduce your development time and costs when setting up or monitoring a serial link by using SuperMonitor.

SuperMonitor is able to execute not only simple short measurements but also long day and night runs. In Windows\* up to two measurements can run simultaneously using the Twin version.

PCMCIA option for Notebooks, available

ADONTEC Ltd

Phone: +49-7043-9000-20

FAX: +49-7043-9000-21

[www.adontec.com](http://www.adontec.com)

International

Phone: +1-2( )

FAX: +1-2( )

W

In Sweden, call: Phone: +46 ( ) 8

FAX: +46 ( ) 5

In Netherlands, call: Phone: +31 2


FAX: +31 2

N

## Notice To Our Subscribers

Occasionally, *Windows Developer's Journal* makes its mailing list available to vendors of products we think our readers will find interesting. Current subscribers receive free information in the mail from these vendors.

If you prefer that your name not be used in these mailings, please let us know. Just copy or clip this form and send it with your name and address to:



**P.O. Box 56565**

**Boulder, CO 80322**