○ Turbo HAMLOG/Win とのデータのやりとりについて（Ver5.27c以上）

WM_COPYDATA というウインドウメッセージを使って、あなたの作成するアプリケーションからTurbo HAMLOG/Winの入力ウインドウに文字列を送ったり、逆に入力ウインドウの文字列を得ることができます。

```
// ********** Delphi（PASCAL言語）での例 **** Delphi 3.1で実験済み
const
    THW_ENTER = $10000;  // データ送信後、ENTERキーを押したのと同じ
    THW_FOCUS = $20000;  // データ送信後、編集ボックスにフォーカス
    THW_SAVEBOX_ON = $40000;    // データ保存時、確認MessageBox表示あり
    THW_SAVEBOX_OFF = $80000;   // データ保存時、確認MessageBox表示なし
    THW_APPLIHWND  = $100000;   // メインウインドウのハンドルを返す
    THW_RMKS_RIGHT = 32;        // Remarks文字列の右側に追加(v5.09)
    THW_RMKS_LEFT = 64;         // Remarks文字列の左側に挿入(v5.09)
    THW_SHUUSEI_WIN = $200000;  // 修正ウインドウだった場合、確認を表示(v5.09a)
    THW_SHUUSEIWIN4 = $400000;  // 修正ウインドウから取り込む(v5.09a)
    THW_QTH__MST  =  $800000;   // dwData=15の場合(v5.23b)のみ利用可で、
                                // HAMLOGが使用中のMSTからQTHを取り出す。
    THW_DX_ENTITY = $1000000;   // DXエンティティ選択ウインドウを表示させる
(v5.27c進捗版)
procedure TForm1.Button1Click(Sender: TObject);
var
    cmmd: Integer;
    Hwnd1, Hwnd2: THandle;
    cds: TCOPYDATASTRUCT;
    buffs: array [0..255] of Char;
begin
    Hwnd1 := FindWindow('TThwin', nil);
    if Hwnd1 < 1 then   //      ハムログが起動していない
        Exit;
    cmmd := 1;  //      コールサインを指定
    cmmd := cmmd or THW_FOCUS;
    cmmd := cmmd or THW_ENTER;
    cds.dwData := cmmd;
    StrPCopy(buffs, Edit1.Text);  // コールサイン文字列
    cds.cbData := StrLen(buffs) + 1;  // 文字列の長さ(+1がいいみたい)
    cds.lpData := @buffs[0];      // 文字列のポインタ
    Hwnd2 := SendMessage(Hwnd1, WM_COPYDATA, Form1.Handle, LPARAM(@cds));
    if Hwnd2 > 0 then
        SetForegroundWindow(Hwnd2);  // 入力ウインドウにフォーカス
end;

/************ C言語での例 ****** 実験してない *********/
#define THW_ENTER  0x10000
#define THW_FOCUS  0x20000
#define THW_SAVEBOX_ON  0x40000 // データ保存時、確認MessageBox表示あり
#define THW_SAVEBOX_OFF 0x80000 // データ保存時、確認MessageBox表示なし
#define THW_APPLIHWND  0x100000
#define THW_RMKS_RIGHT 32
#define THW_RMKS_LEFT  64
$define THW_SHUUSEI_WIN 0x200000
$define THW_SHUUSEIWIN4 0x400000
#define THW_QTH__MS 0x800000
#define THW_DX_ENTITY 0x1000000 // DXエンティティ選択ウインドウを表示させる

{
    int  cmmd;
```

```
    HWND  Hwnd1, Hwnd2;
    COPYDATASTRUCT  cds;
    char  buffs[256];
    Hwnd1 = FindWindow("TThwin", NULL);
    if (Hwnd1 < 1)      //      ハムログが起動していない
        return;
    cmmd = 1;   //      コールサインを指定
    cmmd |= THW_FOCUS;
    cmmd |= THW_ENTER;
    cds.dwData = cmmd;
    strcpy(buffs, "JG1MOU");      // コールサイン文字列
    cds.cbData = strlen(buffs) + 1; // 文字列の長さ(ヌル文字分を+1)
    cds.lpData = &buffs[0];       // 文字列のポインタ
    Hwnd2 = SendMessage(Hwnd1, WM_COPYDATA,
                (WPARAM)Form1->Handle, (LPARAM)&cds);
    if (Hwnd2 > 0)
        SetForegroundWindow(Hwnd2);  // 入力ウインドウにフォーカス
}

' ******* Visual Basicでの例 (Visual Basic Ver6.0評価版で実験済み)
Type COPYDATASTRUCT
    dwData As Long
    cbData As Long
    lpData As String   '  void * のつもり
End Type
Public Const THW_ENTER = &H10000  ' データ送信後、ENTERキーを押したのと同じ
Public Const THW_FOCUS = &H20000  ' データ送信後、編集ボックスにフォーカス
Public Const THW_SAVEBOX_ON = &H40000 ' データ保存時、確認MessageBox表示あり
Public Const THW_SAVEBOX_OFF = &H80000 ' データ保存時、確認MessageBox表示なし
Public Const THW_APPLIHWND  = &H100000 ' メインウインドウのハンドルを返す
Public Const THW_RMKS_RIGHT = 32    ' Remarks文字列の右側に追加(v5.09)
Public Const THW_RMKS_LEFT = 64     ' Remarks文字列の左側に挿入(v5.09)
Public Const THW_SHUUSEI_WIN = &H200000 ' 修正ウインドウだった場合、確認を表示
(v5.09a)
Public Const THW_SHUUSEIWIN4 = &H400000
Public Const THW_DX_ENTITY =  &H1000000

' --- Win32 API --- Visual Basic 6.0 評価版には無かった(^_^;)
Public Const WM_COPYDATA = &H4A
Declare Function FindWindow Lib "user32" Alias "FindWindowA" _
        (ByVal cName As String, ByVal wName As Long) As Long
Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
        (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, _
        lParam As Long) As Long
Declare Function SetForegroundWindow Lib "user32" (wnd As Long) As Long

Private Sub Command1_Click()
Dim cmmd As Long
Dim Hwnd1, Hwnd2 As Long
Dim cds As COPYDATASTRUCT
Dim cbuff As String * 256
    Hwnd1 = FindWindow("TThwin", 0)
    If Hwnd1 < 1 Then   ' ハムログが起動していない
        Exit Sub
    End If
    cmmd = 115  ' 全データ取得
    cmmd = cmmd Or THW_SHUUSEIWIN4 ' 修正ウインドウから取り込む
    cds.dwData = cmmd
```

```
    cbuff = Text1.Text    ' コールサイン文字列をコピー
    cds.cbData = 0
    cds.lpData = 0
    SendMessage(Hwnd1, WM_COPYDATA, hwnd, cds)
End Sub
// ******************************************
```

＊ FindWindow() の左から一つ目のパラメータについて

"TThwin" という文字列を渡す。

＊ FindWindow() の二つ目のパラメータについて

Turbo HAMLOG/Winのメインウインドウのタイトル文字列を渡す。
（正確には、文字列へのポインタを渡します。）これは、Turbo HAMLOG/Winのバージョンによって、また環境設定１で指定したタイトルによって異なります。
通常は、NULLでＯＫです。（Ｃ言語では NULL、Delphiでは nil）

＊ SendMessage()の左から一つ目のパラメータについて
FindWindow()で見つけた、Turbo HAMLOG/Winのハンドルを渡す。

＊ SendMessage()の二つ目のパラメータについて
WM_COPYDATA とする。

＊ SendMessage()の三つ目のパラメータについて
あなたが作成する Turbo HAMLOG/Winを呼び出すアプリケーションの、ウインドウハンドルを渡す。または、Turbo HAMLOG/Winからデータを得たいウインドウのハンドルを渡す。

＊ SendMessage()の四つ目のパラメータについて
COPYDATASTRUCT 構造体で宣言した変数のアドレスを渡す。
LPARAM 型にキャストしないと、コンパイルエラーになるかも。

＊ SendMessage()の戻り値について
Turbo HAMLOG/Winの入力ウインドウのハンドルを返す。（コマンド１７を除く。）
コマンドが不正のときは、ゼロが返る。
このウインドウハンドルで SetForegroundWindow ＡＰＩを呼び出せば、入力ウインドウを前面に表示させることができる。

＊ COPYDATASTRUCT 構造体メンバー dwData について

●コマンドを数値で指定し、Turbo HAMLOG/Winの入力ウインドウに文字列を送る。
1 = コールサインに文字列を送る。
2 = 日付に文字列を送る。
・・1～14まで、Turbo HAMLOG/Winの入力欄の並びのとおり・・
12 = ＱＴＨに文字列を送る。
13 = Remarks１に文字列を送る。
14 = Remarks２に文字列を送る。
15 = コールサインからRemarks２までとチェックボックスが、それぞれ改行された16行のテキストとして送る。 ただし、空改行の欄は転送されない。
16 = ハムログの入力バッファをクリアする。ハムログ内部では、SendMessage()の三つ目のパラメータを引数にして、SetForegroundWindow()を呼び出す。
アプリケーション側では、SendMessage()の前にSetForegroundWindow()でハムログを前面にしておいたほうが良いかもしれない。
17 = デュプチェックのみ行う。ハムログの入力ウインドウでＦ３キーを押したのと同じ。このコマンドでは、SendMessage()の戻り値は交信履歴ウインドウ

のハンドルを返す。
18 = 入力ウインドウの[Save]ボタンをクリックしたのと同じ。このとき dwDataに THW_SAVEBOX_ON と or をとっておけば、Turbo HAMLOG/Winの環境設定にかかわらず「データを登録してよろしいですか？」のMessageBox()が表示される。THW_SAVEBOX_OFF と or をとっておけば、環境設定にかかわらず、データは即登録される。両方ともorした場合は、THW_SAVEBOX_OFFが優先。
19 = 入力ウインドウのCode欄で↓キーを押したのと同じ。(Ver4.52dから)
アプリケーション側では、SendMessage()の前にSetForegroundWindow()でハムログを前面にしておいたほうが良いかもしれない。
20 = 指定したレコード番号の修正ウインドウを表示させます。
21 = 修正ウインドウにデータを送ります。15番と同じ。
22 = PTTのon/offコマンドをTurbo HAMLOG/Winに送らせます。呼び出すたびにon/off
23 = PTTをONにするコマンドをTurbo HAMLOG/Winに送らせます。
24 = PTTのOFFにするコマンドをTurbo HAMLOG/Winに送らせます。
25 = 文字列の先頭バイトが1又は'1'　入力ウインドウのCQのチェックボックスをON
文字列の先頭バイトが2又は'2'　入力ウインドウの1のチェックボックスをON
文字列の先頭バイトが3又は'3'　入力ウインドウの2のチェックボックスをON
文字列の先頭バイトが上記以外　入力ウインドウのDXのチェックボックスをON
(v5.24b)
26 = 文字列の先頭バイトが1又は'1'　入力ウインドウのCQのチェックボックスをOFF
文字列の先頭バイトが2又は'2'　入力ウインドウの1のチェックボックスをOFF
文字列の先頭バイトが3又は'3'　入力ウインドウの2のチェックボックスをOFF
文字列の先頭バイトが上記以外　入力ウインドウのDXのチェックボックスをOFF
(v5.24b)
27 = QSOデータをクローズさせます。(v5.21)この間に、QSOデータの読み書きができます。処理が終わったら、直ちに28を実行してください
28 = QSOデータをオープンさせます。(v5.21)
30 = 文字列として'1'を送る　以後DXエンティティ自動入力をしない。(v5.24c)
文字列として'2'を送る　上記を元に戻す。
文字列として'3'を送る　HAMLOGがアイコン化する
文字列として'4'を送る　上記を元に戻す。

dwData が 1～15 の場合、次の値と or をとることができる。
(Delphi)
```
const THW_ENTER = $10000;  // データ送信後、ENTERキーを押したのと同じ
      THW_FOCUS = $20000;  // データ送信後、編集ボックスにフォーカス
```
(Visual BASIC)
```
Const THW_ENTER = &H10000  ' データ送信後、ENTERキーを押したのと同じ
Const THW_FOCUS = &H20000  ' データ送信後、編集ボックスにフォーカス
```

dwData が 13～15 の場合、次の値と or をとることができる。
(Delphi)
```
const THW_RMKS_RIGHT = 32; // Remarks文字列の右側に追加(v5.09)
      THW_RMKS_LEFT = 64;  // Remarks文字列の左側に挿入(v5.09)
```
(Visual BASIC)
```
Const THW_RMKS_RIGHT = 32  ' Remarks文字列の右側に追加(v5.09)
Const THW_RMKS_LEFT = 64   ' Remarks文字列の左側に挿入(v5.09)
```

dwData が 15 の場合、次の値と or をとることができる。
(Delphi)
```
const THW_QTH_MST = $800000; // MSTのQTHを利用する。(v5.23b)
```
(Visual BASIC)
```
Const THW_QTH_MST = &H800000 ' MSTのQTHを利用する。(v5.23b)
```
●コマンドを数値で指定し、Turbo HAMLOG/Winの入力ウインドウから編集文字列を得る。
101 = コールサインを得る。
102 = 日付
・・101～114まで、Turbo HAMLOG/Winの入力欄の並びのとおり・・
112 = ＱＴＨ

113 = Remarks１
114 = Remarks２
115 = コールサインからRemarks２までが、それぞれ改行されたテキストとして
得ることができる。１行目は空改行。
116 = HAMLOG.HDBファイルのレコード件数とフルパスを得る。
117 = 入力ウインドウの方位と距離の文字列を得る。
118 = メインウインドウの選択行の、コールサインからRemarks２までが、それぞれ
改行されたテキストとして得ることができる。(v5.20b)
119 = lpDataにレコード番号の文字列へのポインタを渡すと、そのレコード番号の
データの、コールサインからRemarks２までが、それぞれ改行されたテキス
トとして得ることができる。指定レコード番号が範囲外の場合は、最終レ
コード番号のデータ。(v5.20b)

dwData が 101～117 の場合、THW_APPLIHWND と or をとると、Turbo HAMLOG/Win のメインウインドウのハンドルを第３引数にSendMessage() が呼び出される。
通常は、入力ウインドウのハンドル。

入力ウインドウが空欄だった場合、呼び出し側アプリの dwData はゼロとなり、空欄でなければ１となる。

ハムログ内部では、(cbData==0)または(lpData==NULL)で呼び出された場合、呼び出し側アプリのSendMessage()の第３引数のハンドルに文字列を返送する。
cbData と lpDataがセットされて呼び出された場合、ハムログ内部では FindWindow により見つけたハンドルに文字列を返送する。通常は前者の方が楽です。

＊ COPYDATASTRUCT 構造体メンバー cbData について

転送するデータの長さ（バイト長）を指定する。
※ dwData が101～115の場合は、cbDataはゼロ、または呼び出し側アプリケーションのウインドウクラスの文字列の長さ指定する。

＊ COPYDATASTRUCT 構造体メンバー lpData について

転送するデータのポインタ（アドレス）を渡す。
※ dwData が101～115の場合は、lpDataには NULL または呼び出し側アプリケーションのウインドウクラスの文字列へのポインタを渡す。[例] "TForm1"

以上　de JG1MOU