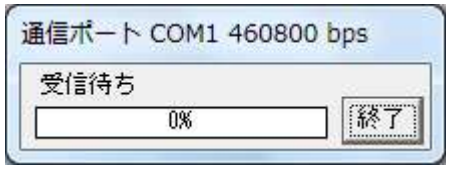
通信プログラム BhtYmText.dll
取扱説明書　第１１版



２０１６年　８月　１８日

コンピュータ・アシスト株式会社

目次

１．プログラム概略

本プログラムは、ＹＭｏｄｅｍプロトコルにより、デンソーウェーブのハンディターミナルへのデータ／プログラム送信、または、ハンディターミナルからデータを受信してテキストファイルを作成するファイル転送のＤＬＬプログラムです。

ＣＳＶファイルでの受信，受信データの追加書き込み，連続受信をサポートしています。

動作するＯＳは、ＷｉｎｄｏｗｓＶｉｓｔａ（３２ｂｉｔ）／７（３２ｂｉｔ／６４ｂｉｔ）／８（３２ｂｉｔ／６４ｂｉｔ）／８．１（３２ｂｉｔ／６４ｂｉｔ）／１０（３２ｂｉｔ／６４ｂｉｔ）

２．実行時の通信状況表示画面

３．関数の使用順序

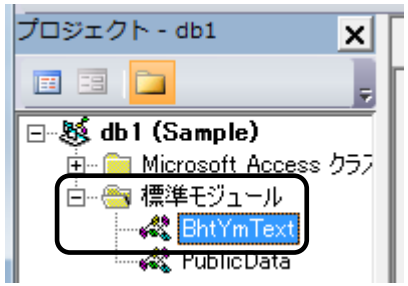
① 通信プログラムを初期化(必ず最初に実行してください)（InitBhtYmText）
② 通信状態表示を変更する場合に設定します。（SetVisible）
③ 通信履歴保存を変更する場合に設定します。（SetSaveLog）
④ 通信をオープンします。（OpenCom）
⑤ 通信を実行します。（Receive, Send）（通信を中断する場合 BreakCom を使用する）
⑥ 続けて通信を実行する場合は、⑤ へ，終了する場合は⑦へ。
⑦ 通信をクローズします。(オープンしたポートは必ずクローズしてください)（CloseCom）

４．関数の宣言

BhtYmTextdll の関数を使用するためには、関数を宣言する必要があります。各プログラム言語における関数宣言のファイルを用意してありますので、これを使用して関数の宣言をしてください。

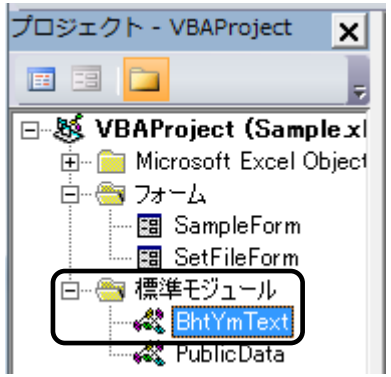
① *Access2010*

[データベ＾スツール]タブ の Visual Basic をクリックして、BhtYmText.bas を「ファイルのインポート(I)」を使って標準モジュールに追加してください。



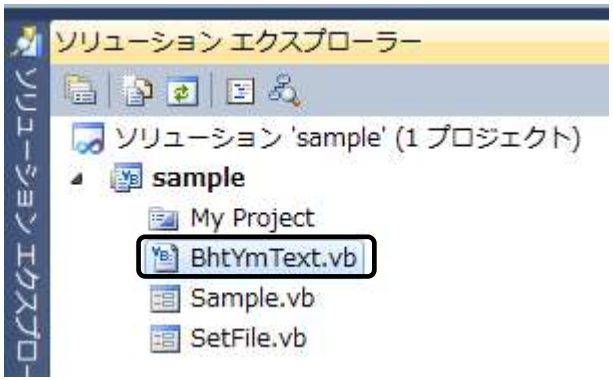
② *Excel2010*

[開発]タブの Visual Basic を クリックして、BhtYmText.bas を「ファイルのインポート(I)」を使って標準モジュールに追加してください。



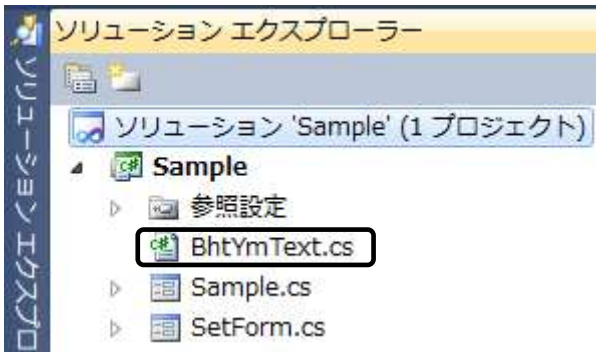
③ *Visual Basic.NET*

BhtYmText.vb をプロジェクトに追加してください。



④ *Visual C#*

BhtYmText.cs をプロジェクトに追加し、
ネームスペースの指定に「using BhtYmTextPrg;」を
追加てください。BhtYmText クラスの関数として使用
してください。



⑤ *Delphi*

BhtYmText.pas をプロジェクトに追加し、Implementation の後に「uses BhtYmText;」を追加してください。
動的にロードする場合は、BhtYmTextD.pas を使用します。（サンプルプログラムを参考にしてください。）

それぞれのサンプルプログラムを参照してください。

５．関数の説明

5-1 InitBhtYmText

【 機能 】

通信プログラムの初期化（必ず最初に呼び出してください！！）
これを実行しないと通信プログラムは動作しません。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

BOOL InitBhtYmText(void);

引数	型	説明
なし		

戻り値	説明
TRUE	正常終了
FALSE	初期化できない

例）

```
if( !InitBhtYmText() )
{
    //プログラムを終了
}
```

② Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003

Function InitBhtYmText() As long

引数	型	説明
なし		

戻り値	説明
0 以外	正常終了
0	初期化できない

例）

```
If InitBhtYmText() = 0 then
    ‘プログラムを終了
End If
```

③ *Visual Basic.NET*

Function InitBhtYmText() As Int32

引数	型	説明
なし		

戻り値	説明
0 以外	正常終了
0	初期化できない

例）

```
If InitBhtYmText() = 0 then
    ‘プログラムを終了
End If
```

④ *Visual C# (BhtYmText クラスの関数)*

Int InitBhtYmText();

引数	型	説明

なし		

戻り値	説明
0 以外	正常終了
0	初期化できない

例）

```
if( BhtYmText.InitBhtYmText() = 0 )
{
    //プログラムを終了
}
```

⑤ *Delphi*

function InitBhtYmText(): LongBool;

引数	型	説明
なし		

戻り値	説明
True	正常終了
False	初期化できない

例）

```
if not InitBhtYmText() then
begin
    //プログラムを終了
end;
```

5-2 GetBhtYmTextVer

【 機能 】

プログラムのバージョンを取得します。初期化されていない場合は取得できません。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

BOOL GetBhtYmTextVer(LPSTR ver);

引数	型	説明
ver	LPSTR	ﾊﾞｰｼﾞｮﾝ情報文字列のポインタ（Ver 1.00 の場合”Ver 1.00”の文字列を返します） ８文字を返しますので NULL を含めて９バイト以上のメモリを確保してください。

戻り値	説明
TRUE	**正常終了**

例）
```
HWND    hWnd;
char    ver[10];  // 10ﾊﾞｲﾄの領域確保

GetBhtYmTextVer( ver );
hWnd = GetActiveWindow();
MessageBox( hWnd, ver, “バージョン表示”, MB_OK | MB_ICONINFORMATION );
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function GetBhtYmTextVer(ByVal Ver As String) As Long

引数	型	説明
Ver	String	ﾊﾞｰｼﾞｮﾝ情報文字列のポインタ（Ver 1.00 の場合”Ver 1.00”の文字列を返します） ８文字を返しますので NULL を含めて９バイト以上のメモリを確保してください。

戻り値	説明
0 以外	**正常終了**

例）　Dim Ver As String

```
Ver = String( 10, vbNillChar )　‘10ﾊﾞｲﾄの領域確保
GetBhtYmTextVer( Ver )
Ver = Left( Ver, InStr( Ver, Chr( 0 ) ) - 1 )　‘文字数確定
MsgBox( Ver, vbOKOnly + vbInformation , “バージョン表示”)
```

③ *Visual Basic.NET*

Function GetBhtYmTextVer(ByVal Ver As String) As Int32

引数	型	説明
Ver	String	ﾊﾞｰｼﾞｮﾝ情報文字列のポインタ（Ver 1.00 の場合”Ver 1.00”の文字列を返します） ８文字を返しますので NULL を含めて９バイト以上のメモリを確保してください。

戻り値	説明
0 以外	**正常終了**

例）　Dim Ver As String

```
Ver = New String( Chr(0), 10 )　‘10ﾊﾞｲﾄの領域確保
GetBhtYmTextVer( Ver )
MessageBox.Show( Ver, “バージョン表示”, MessageBoxButtons.OK, MessageBoxIcon.Information )
```

④ Visual C# (BhtYmText クラスの関数)

Int32 GetBhtYmTextVer(string ver);

引数	型	説明
Ver	string	ﾊﾞｰｼﾞｮﾝ情報文字列のポインタ（Ver 1.00 の場合”Ver 1.00”の文字列を返します） ８文字を返しますので NULL を含めて９バイト以上のメモリを確保してください。

戻り値	説明
0 以外	**正常終了**

例）
```
string  ver;

ver = new string( '¥0', 10 )  //10ﾊﾞｲﾄの領域確保
BhtYmText.GetBhtYmTextVer( ver );
MessageBox.Show( ver, “バージョン表示”, MessageBoxButtons.OK, MessageBoxIcon.Information );
```

⑤ Delphi

function GetBhtYmTextVer(Ver: PAnsiChar): LongBool;

引数	型	説明
Ver	PAnsiChar	ﾊﾞｰｼﾞｮﾝ情報文字列のポインタ（Ver 1.00 の場合”Ver 1.00”の文字列を返します） ８文字を返しますので NULL を含めて９バイト以上のメモリを確保してください。

戻り値	説明
True	**正常終了**

例）
```
var
  Ver: Array[0..9] of AnsiChar;  //10ﾊﾞｲﾄの領域確保

  GetBhtYmTextVer( Ver );
  Application.MessageBox( PChar(String(Ver)),’バージョン表示’, MB_OK + MB_ICONINFORMATION );
```

5-3 OpenCom

【 機能 】

通信ポートをオープンします。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

long OpenCom(long,hWnd, LPCSTR ComPort, long Speed);

引数	型	説明
hWnd	long	ウィンドウハンドル
ComPort	LPCSTR	通信ポート文字列のポインタを指定します。 （COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8）
Speed	long	通信スピードを指定します。 9600, 19200, 38400, 57600, 115200, 460800　これ以外の場合は 115200 でオープンします。

戻り値	説明
-1 以外	**正常終了 通信ハンドルを返します。**
-1	**オープンできない**

例）
```
HWND    hWnd;
long    hCom, Speed;
Char    Comport[6];

hWnd = GetActiveWindow();
strcpy( Comport,” COM1” );
Speed = 115200;
hCom = OpenCom( (long)hWnd, Comport, Speed );
if(hCom = -1)
{
  //異常終了処理
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function OpenCom(ByVal hWnd As Long, ByVal ComPort As String, ByVal Speed As Long) As Long

引数	型	説明
hWnd	Long	ウィンドウハンドル
ComPort	String	通信ポート文字列を指定します。 （COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8）
Speed	Long	通信スピードを指定します。 9600, 19200, 38400, 57600, 115200　これ以外の場合は 115200 でオープンします。

戻り値	説明
-1 以外	**正常終了 通信ハンドルを返します。**
-1	**オープンできない**

例）
```
Dim hWnd As Long                'Excel の場合
Dim hCom As Long
Dim Speed As Long
Dim ComPort As String

hWnd = GetActiveWindow();    'Excel の場合
ComPort = " COM1"
Speed = 115200
hCom = OpenCom( hWnd, ComPort, Speed )
If hCom = -1 Then

    '異常終了処理

End If
```

③ *Visual Basic.NET*

Function OpenCom(ByVal hWnd As Int32, ByVal ComPort As String, ByVal Speed As Int32) As Int32

引数	型	説明
hWnd	Long	ウィンドウハンドル
ComPort	String	通信ポート文字列を指定します。 （COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8）
Speed	Long	通信スピードを指定します。 9600, 19200, 38400, 57600, 115200　これ以外の場合は 115200 でオープンします。

戻り値	説明
-1 以外	**正常終了 通信ハンドルを返します。**
-1	**オープンできない**

例）
```
Dim hCom As Long
Dim Speed As Long
Dim Comport As String

Comport = ” COM1”
Speed = 115200
hCom = OpenCom( Me.Handle().ToInt32, Comport, Speed )
If hCom = -1 Then

    ‘異常終了処理

End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 OpenCom(Int32 hWnd, String ComPort, Int32 Speed);

引数	型	説明
hWnd	Int32	ウィンドウハンドル
ComPort	String	通信ポート文字列を指定します。 （COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8）
Speed	Int32	通信スピードを指定します。 9600, 19200, 38400, 57600, 115200　これ以外の場合は 115200 でオープンします。

戻り値	説明
-1 以外	**正常終了 通信ハンドルを返します。**
-1	**オープンできない**

例）

```
Int32   hCom, Speed;
string  Comport;

Comport = ” COM1” ;
Speed = 115200;
hCom = BhtYmText.OpenCom( this.Handle.ToInt32(), Comport, Speed );
if( hCom = -1 )
{

  //異常終了処理

}
```

⑤ *Delphi 言語*

function OpenCom(hWnd: Longint; ComPort; PAnsiChar; Speed: Longint): Longint;

引数	型	説明
hWnd	Longint	Windows ウィンドウハンドル
ComPort	PAnsiChar	通信ポート文字列のポインタを指定します。 （COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8）
Speed	Longint	通信スピードを指定します。 9600, 19200, 38400, 57600, 115200　これ以外の場合は 115200 でオープンします。

戻り値	説明
-1 以外	**正常終了 通信ハンドルを返します。**
-1	**オープンできない**

例）
```
hCom, Speed: Longint;
Comport:     AnsiString;

Comport := ‘COM1’ ;
Speed := 115200;
hCom := OpenCom( Longint(Handle), PAnsiChar(comport), Speed );
if hCom = -1 then
begin

  //異常終了処理

end;
```

5-4 SetVisible

【 機能 】

通信状態の表示／非表示を設定します。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

BOOL SetVisible(long v, long s);

引数	型	説明
v	long	0 = 非表示　0 以外 = 表示
s	long	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
TRUE	正常終了
FALSE	設定変更失敗（初期化されていない）

例）

```
if( SetVisible( 0, 1 ) )
{
  //非表示に変更し保存
}
else
{
  //変更失敗
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function SetVisible(ByVal v As Long, ByVal s As Long) As Long

引数	型	説明
v	Long	0 = 非表示　0 以外 = 表示
s	Long	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0 以外	**正常終了**
0	**設定変更失敗（初期化されていない）**

例）

```
If SetVisible( 0 ,1 ) <> 0 Then
    ‘非表示に変更し保存
Else
    ‘変更失敗
End If
```

③ *Visual Basic.NET*

Function SetVisible(ByVal v As Int32, ByVal s As Int32) As Int32

引数	型	説明
v	Int32	0 = 非表示　0 以外 = 表示
s	Int32	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0 以外	**正常終了**
0	**設定変更失敗（初期化されていない）**

例）

```
If SetVisible( 0, 1 ) <> 0 Then
    ‘非表示に変更し保存
Else
    ‘変更失敗
End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 SetVisible(Int32 v, Int32 s);

引数	型	説明
v	Int32	0 = 非表示　0 以外 = 表示
s	Int32	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0 以外	**正常終了**
0	**設定変更失敗（初期化されていない）**

例）

```
if( BhtYmText.SetVisible( 0, 1 ) != 0 )
{
  //非表示に変更し保存
}
else
{
  //変更失敗
}
```

⑤ *Delphi 言語*

fuction SetVisible(v, s: Longint): LongBool;

引数	型	説明
v	Longint	0 = 非表示　0 以外 = 表示
s	Longint	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
True	**正常終了**
False	**設定変更失敗（初期化されていない）**

例）

```
If SetVisible( 0, 1 ) then
begin
  //非表示に変更し保存
end
else
begin
  //変更失敗
End;
```

5-5 SetSaveLog

【 機能 】

通信履歴保存のＯＮ／ＯＦＦ，日／月／年ごとの保存，保存フォルダを設定します。

通信履歴保存ファイル名は次のようになります。

日ごとに保存ファイル名： BYTCOM**_YYYY_MM_DD.log
月ごとに保存ファイル名： BYTCOM**_YYYY_MM.log
年ごとに保存ファイル名： BYTCOM**_YYYY.log
** ： 通信ポート No

通信履歴ファイルのデータは次のように保存されます。

01/01 00:00:00	TANA.CSV	100 件	受信	正常終了
01/01 00:01:00	TANA.CSV	100 件	受信	中断
01/01 00:02:00	TANA.CSV	100 件	受信	タイムアウト
01/01 00:10:00	MASTER.CSV	10000 件	送信	正常終了
01/02 02:00:00	ASTB30Q.PD4	358900 ﾊﾞｲﾄ	送信	中断
01/02 02:10:00	ASTB30Q.PD4	358900 ﾊﾞｲﾄ	送信	タイムアウト
01/02 02:20:00	ASTB30Q.PD4	358900 ﾊﾞｲﾄ	送信	正常終了

通信日時　　ファイル名　　通信方向　　通信結果

ﾃﾞｰﾀﾌｧｲﾙの場合：ﾃﾞｰﾀ件数
ﾌﾟﾛｸﾞﾗﾑの場合　：ﾊﾞｲﾄ数

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

long SetSaveLog(long le, long lfm, LPCSTR Folder, long s);

引数	型	説明
le	long	0　　= 通信履歴を保存しない(この場合 lfm と Folder を無視します。) 0 以外 = 通信履歴を保存する
lfm	long	0 = 日ごとに保存(ファイル名：BYTCOM**_YYYY_MM_DD.log)　　** ： 通信ポート No 1 = 月ごとに保存(ファイル名：BYTCOM**_YYYY_MM.log) 2 = 年ごとに保存(ファイル名：BYTCOM**_YYYY.log)
Folder	LPCSTR	通信履歴を保存するフォルダ文字列のポインタ(NULL 文字列の場合はカレントフォルダを設定)
s	long	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**lfm の設定が不良**
3	**設定 Folder が存在しない**

例)　long　ret;

ret = SetSaveLog(1, 1, ””, 1);　//通信履歴を月ごとにカレントフォルダに保存を設定し保存する

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function SetSaveLog(ByVal le As Long, ByVal lfm As Long, ByVal Folder As String, _
ByVal s As Long) As Long

引数	型	説明
le	Long	0 = 通信履歴を保存しない(この場合 lfm と Folder を無視します。) 0 以外 = 通信履歴を保存する
lfm	Long	0 = 日ごとに保存(ファイル名 : BYTCOM**_YYYY_MM_DD.log) ** : 通信ポート No 1 = 月ごとに保存(ファイル名 : BYTCOM**_YYYY_MM.log) 2 = 年ごとに保存(ファイル名 : BYTCOM**_YYYY.log)
Folder	String	通信履歴を保存するフォルダ文字列(NULL 文字列の場合はカレントフォルダを設定)
s	Long	0 = 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗 (初期化されていない)**
2	**lfm の設定が不良**
3	**設定 Folder が存在しない**

例) Dim ret As Long

ret = SetSaveLog(1, 1, ”” , 1) ‘ 通信履歴を月ごとにカレントフォルダに保存を設定し保存する

③ *Visual Basic.NET*

Function SetSaveLog(ByVal le As Int32, ByVal lfm As Int32, ByVal Folder As String, _
ByVal s As Int32) As Int32

引数	型	説明
le	Int32	0　　= 通信履歴を保存しない(この場合 lfm と Folder を無視します。) 0 以外 = 通信履歴を保存する
lfm	Int32	0 = 日ごとに保存(ファイル名 : BYTCOM**_YYYY_MM_DD.log)　　** : 通信ポート No 1 = 月ごとに保存(ファイル名 : BYTCOM**_YYYY_MM.log) 2 = 年ごとに保存(ファイル名 : BYTCOM**_YYYY.log)
Folder	String	通信履歴を保存するフォルダ文字列(NULL 文字列の場合はカレントフォルダを設定)
s	Int32	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**lfm の設定が不良**
3	**設定 Folder が存在しない**

例)　Dim ret As Int32

ret = SetSaveLog(1, 1, ”” , 1)　‘ 通信履歴を月ごとにカレントフォルダに保存を設定し保存する

④ Visual C＃（BhtYmText クラスの関数）

Int32 SetSaveLog(Int32 le, Int32 lfm, string Folder, Int32 s);

引数	型	説明
le	Int32	0　　= 通信履歴を保存しない(この場合 lfm と Folder を無視します。) 0 以外 = 通信履歴を保存する
lfm	Int32	0 = 日ごとに保存(ファイル名：BYTCOM**_YYYY_MM_DD.log)　　** ： 通信ポート No 1 = 月ごとに保存(ファイル名：BYTCOM**_YYYY_MM.log) 2 = 年ごとに保存(ファイル名：BYTCOM**_YYYY.log)
Folder	String	通信履歴を保存するフォルダ文字列(NULL 文字列の場合はカレントフォルダを設定)
s	Int32	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**lfm の設定が不良**
3	**設定 Folder が存在しない**

例）　Int32　ret;

ret = BhtYmText.SetSaveLog(1, 1, ””, 1);　// 通信履歴を月ごとにカレントフォルダに保存を設定し保存する

⑤ *Delphi 言語*

function SetSaveLog(le, lfm: Longint; Folder: PAnsiChar; s: Longint): Longint;

引数	型	説明
le	Longint	0　　= 通信履歴を保存しない(この場合 lfm と Folder を無視します。) 0 以外 = 通信履歴を保存する
lfm	Longint	0 = 日ごとに保存(ファイル名：BYTCOM**_YYYY_MM_DD.log)　　** ： 通信ポート No 1 = 月ごとに保存(ファイル名：BYTCOM**_YYYY_MM.log) 2 = 年ごとに保存(ファイル名：BYTCOM**_YYYY.log)
Folder	PAnsiChar	通信履歴を保存するフォルダ文字列のポインタ(NULL 文字列の場合はカレントフォルダを設定)
s	Longint	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**lfm の設定が不良**
3	**設定 Folder が存在しない**

例）
```
var
    ret:  Longint;

    ret := SetSaveLog( 1, 1, '', 1 );  // 通信履歴を月ごとにカレントフォルダに保存を設定し保存する
```

5-6 SetDefFolder

【 機能 】

受信／送信ファイルのデフォルトフォルダを設定します。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

long SetDefFolder(LPCSTR DefFolder, long m, long s);

引数	型	説明
DefFolder	LPCSTR	受信/送信ファイルのデフォルトフォルダのポインタ
m	long	0 = 受信デフォルトフォルダ設定 0 以外 = 送信デフォルトフォルダ設定
s	long	0 = 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**設定 Folder が存在しない**

例） long ret;

ret = SetDefFolder(“c:¥¥MyFolder¥¥RxFolder”, 0, 1);

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function SetDefFolder(ByVal DefFolder As String, ByVal m As Long, ByVal s As Long) As Long

引数	型	説明
DefFolder	String	受信/送信ファイルのデフォルトフォルダのポインタ
m	long	0　　= 受信デフォルトフォルダ設定 0 以外 = 送信デフォルトフォルダ設定
s	long	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**設定 Folder が存在しない**

例）　Dim ret As Long

　　ret = SetDefFolder(“c:¥MyFolder¥RxFolder”, 0, 1)

③ *Visual Basic.NET*

Function SetDefFolder(ByVal DefFolder As String, ByVal m As Int32, ByVal s As Int32) As Int32

引数	型	説明
DefFolder	String	受信/送信ファイルのデフォルトフォルダのポインタ
m	long	0　　= 受信デフォルトフォルダ設定 0 以外 = 送信デフォルトフォルダ設定
s	long	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**設定 Folder が存在しない**

例）　Dim ret As Int32

　　ret = SetDefFolder(“c:¥MyFolder¥RxFolder”, 0, 1)

④ Visual C＃（BhtYmText クラスの関数）

Int32 SetDefFolder(string DefFolder, Int32 m, Int32 s);

引数	型	説明
DefFolder	string	受信/送信ファイルのデフォルトフォルダのポインタ
m	Int32	0　　= 受信デフォルトフォルダ設定 0 以外 = 送信デフォルトフォルダ設定
s	Int32	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**設定 Folder が存在しない**

例）　Int32　ret;

ret = BhtYmText.SetDefFolder(“c:¥¥MyFolder¥¥RxFolder”, 0, 1);

⑤ *Delphi 言語*

function SetDefFolder(DefFolder: PAnsiChar; m,s: Longint): Longint;

引数	型	説明
DefFolder	PAnsiChar	受信/送信ファイルのデフォルトフォルダのポインタ
m	Longint	0　　= 受信デフォルトフォルダ設定 0 以外 = 送信デフォルトフォルダ設定
s	Longint	0　　= 一時的に設定を変更します。 0 以外 = 設定を変更して保存します。

戻り値	説明
0	**正常終了**
1	**設定変更失敗（初期化されていない）**
2	**設定 Folder が存在しない**

例）
```
var
    ret:  Longint;

    ret := SetDefFolder( ‘c:¥MyFolder¥RxFolder’, 0, 1 );
```

5-7 GetDefFolder

【 機能 】

受信／送信ファイルのデフォルトフォルダを取得します。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

BOOL GetDefFolder(long m, LPCSTR DefFolder);

引数	型	説明
m	long	0　　= 受信デフォルトフォルダ取得 0 以外 = 送信デフォルトフォルダ取得
DefFolder	LPCSTR	取得する受信/送信ファイルのデフォルトフォルダのポインタ

戻り値	説明
TRUE	正常終了
FALSE	取得失敗（初期化されていない）

例）
```
HWND    hWnd;
char    RcvFolder[256];  // 256バイトの領域確保

if( GetDefFolder( 0, RcvFolder ) )
{
  //受信デフォルトフォルダ取得
  hWnd = GetActiveWindow();
  MessageBox( hWnd, RcvFolder, “受信デフォルトフォルダ”, MB_OK | MB_ICONINFORMATION);
}
else
{
  //取得失敗
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function GetDefFolder(ByVal m As Long, ByVal DefFolder As String) As Long

引数	型	説明
m	Long	0　　= 受信デフォルトフォルダ取得 0 以外 = 送信デフォルトフォルダ取得
DefFolder	String	取得する受信/送信ファイルのデフォルトフォルダ

戻り値	説明
0 以外	**正常終了**
0	**取得失敗（初期化されていない）**

例）

```
Dim RcvFolder As String

RcvFolder = String( 255, vbNillChar )    ‘255ﾊﾞｲﾄの領域確保
If GetDefFolder( 0, RcvFolder ) <> 0 Then
   ‘受信デフォルトフォルダ取得
  RcvFolder = Left( RcvFolder, InStr(RcvFolder, Chr( 0 ) ) - 1 )    ‘文字数確定
  MsgBox(RcvFolder, vbOKOnly + vbInformation , “受信デフォルトフォルダ”)
Else

   ‘取得失敗

End If
```

③ *Visual Basic.NET*

Function GetDefFolder(ByVal m As Int32, ByVal DefFolder As String) As Int32

引数	型	説明
m	Int32	0　　= 受信デフォルトフォルダ取得 0 以外 = 送信デフォルトフォルダ取得
DefFolder	String	取得する受信/送信ファイルのデフォルトフォルダ

戻り値	説明
0 以外	**正常終了**
0	**取得失敗（初期化されていない）**

例）

```
Dim  RcvFolder As String

RcvFolder = New String(Chr(0), 255)    ‘255ﾊﾞｲﾄの領域を確保
If GetDefFolder( 0, RcvFolder ) <> 0 Then

  ‘受信デフォルトフォルダ取得
  MessageBox.Show ( RcvFolder, “受信デフォルトフォルダ”, MessageBoxButtons.OK, MessageBoxIcon.Information)

Else

  ‘取得失敗

End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 GetDefFolder(Int32 m, string DefFolder);

引数	型	説明
m	Int32	0　　= 受信デフォルトフォルダ取得 0 以外 = 送信デフォルトフォルダ取得
DefFolder	string	取得する受信/送信ファイルのデフォルトフォルダ

戻り値	説明
0 以外	**正常終了**
0	**取得失敗（初期化されていない）**

例）

```
string  RcvFolder;

RcvFolder = new string( '¥0', 255);  //255ﾊﾞｲﾄの領域を確保
if( BhtYmText.GetDefFolder( 0, RcvFolder ) )
{
  //受信デフォルトフォルダ取得
  MessageBox.Show ( RcvFolder, "受信デフォルトフォルダ", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{

  //取得失敗

}
```

⑤ *Delphi 言語*

function GetDefFolder(m: Longint; DefFolder: PAnsiChar): LongBool;

引数	型	説明
m	Longint	0　　= 受信デフォルトフォルダ取得 0 以外 = 送信デフォルトフォルダ取得
DefFolder	PAnsiChar	取得する受信/送信ファイルのデフォルトフォルダのポインタ

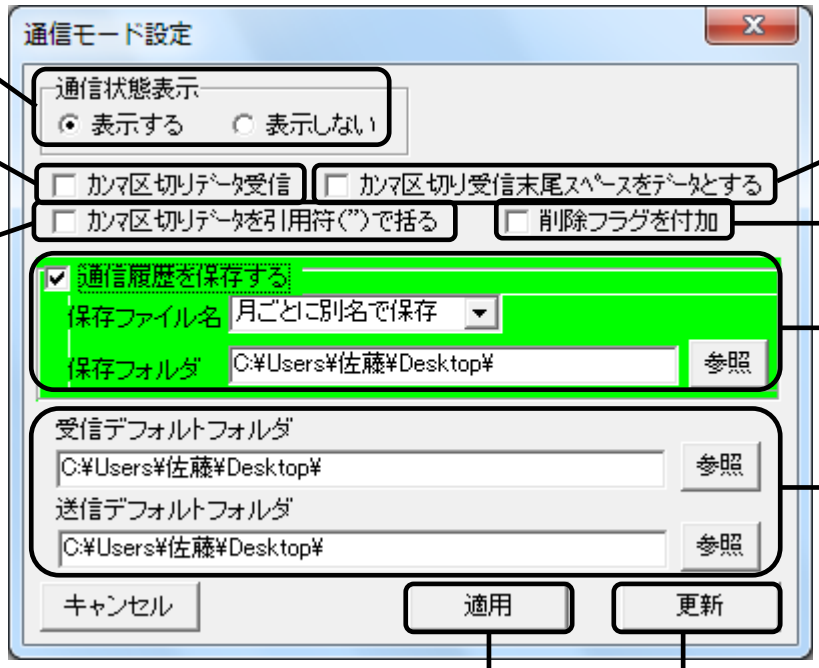
戻り値	説明
True	**正常終了**
False	**取得失敗（初期化されていない）**

例）
```
var
  RcvFolder: Array[0..255] of AnsiChar;  //256ﾊﾞｲﾄの領域を確保

  if GetDefFolder( 0, RcvFolder ) then
  begin
    //受信デフォルトフォルダ取得
    Application.MessageBox( PChar(String(RcvFolder)), '受信デフォルトフォルダ', MB_OK + MB_ICONINFORMATION );
  end
  else
  begin
    //取得失敗

  end;
```

5-8 SetComMode

【 機能 】

通信状態の表示／非表示，通信履歴保存，送信／受信ファイルのデフォルトフォルダを設定するダイアログを表示します。



通信状態の表示

ＣＳＶファイル以外のデータもカンマ区切りデータで受信します。

カンマ区切りデータ受信設定またはＣＳＶファイル受信の場合、カンマ区切りデータを引用符（“）で括りテキストファイルを作成します。

ＣＳＶファイル送信の場合、カンマ区切りデータを引用符（“）で括りテキストファイルを送信します。

カンマ区切りデータ受信の末尾スペースをデータとして保存します.

末尾に削除フラグ１桁を付加します.

通信履歴保存設定

受信／送信デフォルトフォルダ

設定を適用します

設定を適用し保存します

保存する通信履歴ファイル名
日ごとに保存ファイル名： BYTCOM**_YYYY_MM_DD.log
月ごとに保存ファイル名： BYTCOM**_YYYY_MM.log
年ごとに保存ファイル名： BYTCOM**_YYYY.log
** ： 通信ポートNo

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

```
BOOL SetComMode( long hWnd );
```

引数	型	説明
hWnd	long	ウィンドウハンドル

戻り値	説明
TRUE	正常終了
FALSE	設定変更不可（初期化されていない）

例）

```
HWND    hWnd;

hWnd = GetActiveWindow();
if( SetComMode( (long)hWnd ) )
{
  //通信条件設定変更終了

}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function SetComMode(ByVal hWnd As Long) As Long

引数	型	説明
hWnd	Long	ウィンドウハンドル

戻り値	説明
0 以外	正常終了
0	設定変更不可（初期化されていない）

例）
```
Dim ret As Long

ret = SetComMode( hWnd )
```

③ *Visual Basic.NET*

Function SetComMode(ByVal hWnd As Int32) As Int32

引数	型	説明
hWnd	Int32	ウィンドウハンドル

戻り値	説明
0 以外	正常終了
0	設定変更不可（初期化されていない）

例）
```
Dim ret As Int32

ret = SetComMode( Me.Handle().ToInt32 )
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 SetComMode(Int32 hWnd);

引数	型	説明
hWnd	Int32	ウィンドウハンドル

戻り値	説明
0 以外	正常終了
0	設定変更不可（初期化されていない）

例）
```
Int32  ret;

ret = BhtYmText.SetComMode( this.Handle().ToInt32 );
```

⑤ Delphi 言語

function SetComMode(hWnd: Longint): LongBool;

引数	型	説明
hWnd	Longint	ウィンドウハンドル

戻り値	説明
True	正常終了
False	設定変更不可（初期化されていない）

例）
```
b:  LongBool;

b := SetComMode( Longint(Handle) );
```

5-9 BreakCom

【 機能 】

通信中の処理を中断します。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

long BreakCom(long hCom);

引数	型	説明
hCom	long	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	正常終了
1	通信ポートをオープンしていない
2	すでに通信を終了している

例)
```
HWND    hWnd;
long    hCom, ret;

hWnd = GetActiveWindow();
hCom = OpenCom( (long)hWnd, “COM1”, 115200 );
if(hCom != -1)
{
  //通信処理
   ・・・・・・・
ret = BreakCom( hCom );
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function BreakCom(ByVal hCom As Long) As Long

引数	型	説明
hCom	Long	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	**正常終了**
1	**通信ポートをオープンしていない**
2	**すでに通信を終了している**

例)

```
Dim hWnd As Long                ‘Excel の場合
Dim hCom As Long, ret As Long

hWnd = GetActiveWindow();       ‘Excel の場合
hCom = OpenCom( hWnd, “COM1”, 115200 )
If hCom <> -1 Then

   ‘ 通信処理
  ・・・・・・・
  ret = BreakCom( hCom )

End If
```

③ *Visual Basic.NET*

Function BreakCom(ByVal hCom As Int32) As Int32

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	**正常終了**
1	**通信ポートをオープンしていない**
2	**すでに通信を終了している**

例)

```
Dim hCom As Int32, ret As Int32

hCom = OpenCom( Me.Handle().ToInt32,  “COM1”, 115200 )
If hCom <> -1 Then

   ‘ 通信処理
   ・・・・・・・・
  ret = BreakCom( hCom )

End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 BreakCom(Int32 hCom);

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	正常終了
1	通信ポートをオープンしていない
2	すでに通信を終了している

例）

```
Int32  hCom, ret;

hCom = BhtYmText.OpenCom( this.Handle().ToInt32,  “COM1”, 115200 );
if(hCom != -1)
{
  //通信処理
   ・・・・・・・・
  ret = BhtYmText.BreakCom( hCom );
}
```

⑤ *Delphi 言語*

function BreakCom(hCom: Longint): Longint;

引数	型	説明
hCom	Longint	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	**正常終了**
1	**通信ポートをオープンしていない**
2	**すでに通信を終了している**

例)

```
hCom, ret:  Longint;

hCom := OpenCom( Longint(Handle),  'COM1', 115200 );
if hCom <> -1 then
begin
  //通信処理
   ・・・・・・・・
  ret := BreakCom( hCom );
end;
```

5-10 CloseCom

【 機能 】

オープンした通信ポートをクローズします。通信中の場合は、BreakCom で通信を中断してから CloseCom を実行してください。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

long CloseCom(long hCom);

引数	型	説明
hCom	long	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	正常終了
1	通信ポートをオープンしていない
2	通信中です

例)
```
HWND    hWnd;
long    hCom, ret;

hWnd = GetActiveWindow();
hCom = OpenCom( (long)hWnd, “COM1”, 115200 );
if( hCom != -1 )
{
  //通信処理
   ・・・・・・・
  ret = CloseCom(hCom);
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function CloseCom(ByVal hCom As Long) As Long

引数	型	説明
hCom	Long	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	**正常終了**
1	**通信ポートをオープンしていない**
2	**通信中です**

例)
```
Dim hWnd As Long                ‘Excel の場合
Dim hCom As Long, ret As Long

hWnd = GetActiveWindow();       ‘Excel の場合
hCom = OpenCom( hWnd, “COM1”, 115200 )
If hCom <> -1 Then

   ‘ 通信処理
  ・・・・・・・
  ret = CloseCom( hCom )

End If
```

③ *Visual Basic.NET*

Function CloseCom(ByVal hCom As Int32) As Int32

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	**正常終了**
1	**通信ポートをオープンしていない**
2	**通信中です**

例)

```
Dim hCom As Int32, ret As Int32

hCom = OpenCom( Me.Handle().ToInt32,  “COM1”, 115200 )
If hCom <> -1 Then

   ‘ 通信処理
   . . . . . .
  ret = CloseCom( hCom )

End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 CloseCom(Int32 hCom);

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	正常終了
1	通信ポートをオープンしていない
2	通信中です

例）

```
Int32  hCom, ret;

hCom = BhtYmText.OpenCom( this.Handle().ToInt32,  “COM1”, 115200 );
if( hCom != -1 )
{
  //通信処理
   ・・・・・・・・
  ret = BhtYmText.CloseCom( hCom );
}
```

⑤ *Delphi 言語*

function CloseCom(hCom: Longint): Longint;

引数	型	説明
hCom	Longint	通信ハンドル(OpenCom が返した値)

戻り値	説明
0	**正常終了**
1	**通信ポートをオープンしていない**
2	**通信中です**

例)　hCom, ret: Longint;

```
hCom := OpenCom( Longint(Handle),  'COM1', 115200 );
if hCom <> -1 then
begin
  //通信処理
   ・・・・・・・
  ret := CloseCom(hCom);
end;
```

5-11 Send

【 機能 】

オープンした通信ポートにファイルを送信します。送信ファイルは、固定長データファイル，ＣＳＶファイル，ＰＤ４プログラム，ＦＮ３プログラムです。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

long Send(long hCom, LPCSTR SendFileName, LPCSTR BhtFileName, LPCSTR FieldData);

引数	型	説明
hCom	long	通信ハンドル(OpenCom が返した値)
SendFileName	LPCSTR	送信ファイルのポインタ （フォルダを指定しない場合はデフォルトフォルダ内のファイルになります）
BhtFileName	LPCSTR	ハンディターミナル内で使用するファイル名(拡張子を含め１２文字以内)のポインタ 送信ファイルの拡張子と同じ拡張子を指定をしてください。 NULL 文字を指定すると送信ファイル名と同じファイル名になります。
FieldData	LPCSTR	フィールド情報文字列のポインタ（送信ファイルがプログラムファイルの場合は無視されます） 「, 」を区切り記号にして次のように指定してください。 6,13,1 NULL 文字を指定すると、FLD 拡張子のフィールドファイルからフィールド情報を取得する。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
long    hCom, ret;

ret = Send( hCom, “ASTB30B.PD4”, “”, “” );
if( ret == 0 )
{
  //送信終了処理
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function Send(ByVal hCom As Long, ByVal SendFileName As String, ByVal BhtFileName As String, _
ByVal FieldData As String) As Long

引数	型	説明
hCom	Long	通信ハンドル(OpenCom が返した値)
SendFileName	String	送信ファイル（フォルダを指定しない場合はデフォルトフォルダ内のファイルになります）
BhtFileName	String	ハンディターミナル内で使用するファイル名(拡張子を含め１２文字以内) 送信ファイルの拡張子と同じ拡張子を指定をしてください。 NULL 文字を指定すると送信ファイル名と同じファイル名になります。
FieldData	String	フィールド情報文字列（送信ファイルがプログラムファイルの場合は無視されます） 「, 」を区切り記号にして次のように指定してください。 6,13,1 NULL 文字を指定すると、FLD 拡張子のフィールドファイルからフィールド情報を取得する。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例） Dim hCom As Long, ret As Long

```
ret = Send( hCom, “ASTB30B.PD4”, “”, “” )
If ret = 0 Then

   ‘送信終了処理

End If
```

③ *Visual Basic.NET*

Function Send(ByVal hCom As Int32, ByVal SendFileName As String, ByVal BhtFileName As String, _
ByVal FieldData As String) As Int32

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)
SendFileName	String	送信ファイル（フォルダを指定しない場合はデフォルトフォルダ内のファイルになります）
BhtFileName	String	ハンディターミナル内で使用するファイル名(拡張子を含め１２文字以内) 送信ファイルの拡張子と同じ拡張子を指定をしてください。 NULL 文字を指定すると送信ファイル名と同じファイル名になります。
FieldData	String	フィールド情報文字列（送信ファイルがプログラムファイルの場合は無視されます） 「, 」を区切り記号にして次のように指定してください。　6,13,1 NULL 文字を指定すると、FLD 拡張子のフィールドファイルからフィールド情報を取得する。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
Dim    hCom As Int32, ret As Int32

ret = Send( hCom, “ASTB30B.PD4”, “”, “” )
If ret = 0 Then

   ‘送信終了処理

End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 Send(Int32 hCom, string SendFileName, string BhtFileName, string FieldData);

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)
SendFileName	string	送信ファイル（フォルダを指定しない場合はデフォルトフォルダ内のファイルになります）
BhtFileName	string	ハンディターミナル内で使用するファイル名(拡張子を含め１２文字以内) 送信ファイルの拡張子と同じ拡張子を指定をしてください。 NULL 文字を指定すると送信ファイル名と同じファイル名になります。
FieldData	string	フィールド情報文字列（送信ファイルがプログラムファイルの場合は無視されます） 「, 」を区切り記号にして次のように指定してください。　6,13,1 NULL 文字を指定すると、FLD 拡張子のフィールドファイルからフィールド情報を取得する。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）

```
Int32    hCom, ret;

ret = BhtYmText.Send( hCom, “ASTB30B.PD4”, “”, “” );
if( ret == 0 )
{
  //送信終了処理
}
```

⑤ *Delphi 言語*

function Send(hCom: Longint; SendFileName,BhtFileName,FieldData: PAnsiChar): Longint;

引数	型	説明
hCom	Longint	通信ハンドル(OpenCom が返した値)
SendFileName	PAnsiChar	送信ファイル（フォルダを指定しない場合はデフォルトフォルダ内のファイルになります）
BhtFileName	PAnsiChar	ハンディターミナル内で使用するファイル名(拡張子を含め１２文字以内) 送信ファイルの拡張子と同じ拡張子を指定をしてください。 NULL 文字を指定すると送信ファイル名と同じファイル名になります。
FieldData	PAnsiChar	フィールド情報文字列（送信ファイルがプログラムファイルの場合は無視されます） 「, 」を区切り記号にして次のように指定してください。　6,13,1 NULL 文字を指定すると、FLD 拡張子のフィールドファイルからフィールド情報を取得する。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
var
    hCom, ret:  Longint;

    ret := Send( hCom,  'ASTB30B.PD4' , '', '' );
    if( ret = 0 )
    {
      //送信終了処理
    }
```

5-12 Receive

【 機能 】

オープンした通信ポートからファイルを受信します。受信ファイルは、固定長データファイル，ＣＳＶファイル，ＰＤ４プログラム，ＦＮ３プログラムです。ＣＳＶファイルとして受信するのは、ハンディターミナル内のデータファイルの拡張子が「. ＣＳＶ」の場合です。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

```
long Receive( long hCom, LPCSTR FolderName, long RcvMode, long WrMode, LPCSTR RcvFileName,
              LPCSTR FieldData );
```

引数	型	説明
hCom	long	通信ハンドル(OpenCom が返した値)
FolderName	LPCSTR	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	long	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0 = 受信終了で戻ります
WrMode	long	0 以外 = 追加書き込み 0 = 上書き
RcvFileName	LPCSTR	受信ファイル名を保存する領域（15ﾊﾞｲﾄ）を確保し、その領域のポインタ
FieldData	LPCSTR	フィールド情報文字列を保存する領域（256ﾊﾞｲﾄ）を確保し、その領域のポインタ 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） ＰＤ4 プログラム，ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値		説明
	0	正常終了
	0 以外	エラーコード表参照

例）
```
long    hCom, ret;
char    RcvFileName[15];  //15ﾊﾞｲﾄの領域確保
char    FieldData[256];   //256ﾊﾞｲﾄの領域確保

ret = Receive( hCom, “”, 0, 0, RcvFileName, FieldData );
if( ret == 0 )
{
  //受信終了処理
}
else
{
  //受信エラー処理
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function Receive(ByVal hCom As Long, ByVal FolderName As String, ByVal RcvMode As Long, _
ByVal WrMode As Long, ByVal RcvFileName As String, ByVal FieldData As String) As Long

引数	型	説明
hCom	Long	通信ハンドル(OpenCom が返した値)
FolderName	String	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Long	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0 = 受信終了で戻ります
WrMode	Long	0 以外 = 追加書き込み 0 = 上書き
RcvFileName	String	受信ファイル名を保存する領域（15ﾊﾞｲﾄ）を確保
FieldData	String	フィールド情報文字列を保存する領域（255ﾊﾞｲﾄ）を確保 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） ＰＤ4 プログラム, ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
Dim hCom As Long, ret As Long
Dim RcvFileName As String, FieldData As String

RcvFileName = String( 15, vbNillChar )    ‘10ﾊﾞｲﾄの領域確保
FieldData = String( 255, vbNillChar )     ‘255ﾊﾞｲﾄの領域確保
ret = Receive( hCom, “”, 0, 0, RcvFileName, FieldData )
If ret = 0 Then
  RcvFileName = Left( RcvFileName, InStr( RcvFileName, Chr( 0 ) ) - 1 )   ‘文字数確定
  FieldData = Left( FieldData, InStr( FieldData, Chr( 0 ) ) - 1 )         ‘文字数確定
End If
```

③ *Visual Basic.NET*

Function Receive(ByVal hCom As Int32, ByVal FolderName As String, ByVal RcvMode As Int32, _
ByVal WrMode As Int32, ByVal RcvFileName As String, ByVal FieldData As String) As Int32

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)
FolderName	String	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Int32	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0 = 受信終了で戻ります
WrMode	Int32	0 以外 = 追加書き込み 0 = 上書き
RcvFileName	String	受信ファイル名を保存する領域（15ﾊﾞｲﾄ）を確保
FieldData	String	フィールド情報文字列を保存する領域（255ﾊﾞｲﾄ）を確保 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） ＰＤ4 プログラム, ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
Dim hCom As Int32, ret As Int32
Dim RcvFileName As String, FieldData As String

RcvFileName = New String(Chr(0), 15)    ‘15ﾊﾞｲﾄの領域確保
FieldData = New String(Chr(0), 255)     ‘255ﾊﾞｲﾄの領域確保
ret = Receive( hCom, “”, 0, 0, RcvFileName, FieldData )
If ret = 0 Then
   ‘受信終了処理
End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 Receive(Int32 hCom, string FolderName, Int32 RcvMode,Int32 WrMode, string BhtFileName, string FieldData);

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)
FolderName	string	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Int32	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0 = 受信終了で戻ります
WrMode	Int32	0 以外 = 追加書き込み 0 = 上書き
RcvFileName	string	受信ファイル名を保存する領域（15ﾊﾞｲﾄ）を確保
FieldData	string	フィールド情報文字列を保存する領域（255ﾊﾞｲﾄ）を確保 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） ＰＤ4 プログラム, ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
Int32  hCom, ret;
syting RcvFileName, FieldData;

RcvFileName = new String( '¥0' , 15);  //15ﾊﾞｲﾄの領域確保
FieldData = new String( '¥0' , 255);   //255ﾊﾞｲﾄの領域確保
ret = BhtYmText.Receive( hCom,  "" , 0, 0, RcvFileName, FieldData );
if( ret = 0 )
{
  //受信終了処理
}
```

⑤ *Delphi 言語*

function Receive(hCom: Longint; FolderName: PAnsiChar; RcvMode, WrMode: Longint;
RcvFileName, FieldData: PAnsiChar): Longint;

引数	型	説明
hCom	Longint	通信ハンドル(OpenCom が返した値)
FolderName	PAnsiChar	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Longint	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0 = 受信終了で戻ります
WrMode	Longint	0 以外 = 追加書き込み 0 = 上書き
RcvFileName	PAnsiChar	受信ファイル名を保存する領域（15ﾊﾞｲﾄ）を確保し、その領域のポインタ
FieldData	PAnsiChar	フィールド情報文字列を保存する領域（256ﾊﾞｲﾄ）を確保し、その領域のポインタ 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） ＰＤ4 プログラム, ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
var
  hCom, ret:   Longint;
  RcvFileName: Array[0..14] of AnsiChar;   //15ﾊﾞｲﾄの領域確保
  FieldData:   Array[0..255] of AnsiChar;  //256ﾊﾞｲﾄの領域確保

  ret := BhtYmText.Receive( hCom,  '' , 0, 0, RcvFileName, FieldData );
  if ret = 0 then
  begin
    //受信終了処理
  end;
```

5-13 RcvFileChk

【 機能 】

オープンした通信ポートから指定したファイル名とフィールド情報のデータを受信します。指定ファイル名とフィールド情報に一致しない場合は、エラーになります。フィールド情報を指定しない場合はファイル名のみチェックします。受信ファイルは、固定長データファイル，ＣＳＶファイル，ＰＤ４プログラム，ＦＮ３プログラムです。ＣＳＶファイルとして受信するのは、ハンディターミナル内のデータファイルの拡張子が「．ＣＳＶ」の場合です。

【 書式 】

① Visual C++6.0, Visual C++.NET, C++Builder

```
long RcvFileChk( long hCom, LPCSTR FolderName, long RcvMode, long WrMode, LPCSTR RcvFileName,
                 LPCSTR FieldData );
```

引数	型	説明
hCom	long	通信ハンドル(OpenCom が返した値)
FolderName	LPCSTR	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	long	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0　　= 受信終了で戻ります
WrMode	long	0 以外 = 追加書き込み 0　　= 上書き
RcvFileName	LPCSTR	指定する受信ファイル名文字列のポインタ
FieldData	LPCSTR	指定するフィールド情報文字列のポインタ 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） フィールド情報を指定しない場合は、フィールド情報文字列を保存する領域（256ﾊﾞｲﾄ）を確保し、‘¥0’で初期化してその領域のポインタを設定します ＰＤ4 プログラム，ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値		説明
	0	正常終了
	0 以外	エラーコード表参照

例）
```
long    hCom, ret;
char    FieldData[256];   //256ﾊﾞｲﾄの領域確保

memset(FieldData,'¥0',sizeof(FieldData));
ret = Receive( hCom, "", 0, 0, "TANA.DAT", FieldData );
if( ret == 0 )
{
  //受信終了処理
}
else
{
  //受信エラー処理
}
```

② *Visual Basic6.0, Access2000/2002/2003, Excel2000/2002/2003*

Function RcvFileChk(ByVal hCom As Long, ByVal FolderName As String, ByVal RcvMode As Long, _
ByVal WrMode As Long, ByVal RcvFileName As String, ByVal FieldData As String) As Long

引数	型	説明
hCom	Long	通信ハンドル(OpenCom が返した値)
FolderName	String	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Long	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0 = 受信終了で戻ります
WrMode	Long	0 以外 = 追加書き込み 0 = 上書き
RcvFileName	String	指定する受信ファイル名文字列のポインタ
FieldData	String	指定するフィールド情報文字列のポインタ 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） フィールド情報を指定しない場合は、フィールド情報文字列を保存する領域（256ﾊﾞｲﾄ）を確保し、vbNillChar で初期化してその領域のポインタを設定します ＰＤ4 プログラム, ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
Dim hCom As Long, ret As Long
Dim FieldData As String

FieldData = String( 255, vbNillChar )     '255ﾊﾞｲﾄの領域確保
ret = Receive( hCom, "", 0, 0, "TANA.DAT", FieldData )
If ret = 0 Then
  FieldData = Left( FieldData, InStr( FieldData, Chr( 0 ) ) - 1 )          '文字数確定
End If
```

③ *Visual Basic.NET*

Function RcvFileChk(ByVal hCom As Int32, ByVal FolderName As String, ByVal RcvMode As Int32, _
ByVal WrMode As Int32, ByVal RcvFileName As String, ByVal FieldData As String) As Int32

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)
FolderName	String	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Int32	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0　　= 受信終了で戻ります
WrMode	Int32	0 以外 = 追加書き込み 0　　= 上書き
RcvFileName	String	指定する受信ファイル名文字列のポインタ
FieldData	String	指定するフィールド情報文字列のポインタ 「, 」を区切り記号にした文字列になります。（　6,13,1　） フィールド情報を指定しない場合は、フィールド情報文字列を保存する領域（256ﾊﾞｲﾄ）を確保し、 Chr(0)で初期化してその領域のポインタを設定します ＰＤ4 プログラム, ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
Dim hCom As Int32, ret As Int32
Dim FieldData As String

FieldData = New String(Chr(0), 255)     ‘255ﾊﾞｲﾄの領域確保
ret = Receive( hCom, “”, 0, 0, “TANA.DAT”, FieldData )
If ret = 0 Then
   ‘受信終了処理
End If
```

④ Visual C＃（BhtYmText クラスの関数）

Int32 RcvFileChk(Int32 hCom, string FolderName, Int32 RcvMode,Int32 WrMode, string BhtFileName, string FieldData);

引数	型	説明
hCom	Int32	通信ハンドル(OpenCom が返した値)
FolderName	string	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Int32	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0　　= 受信終了で戻ります
WrMode	Int32	0 以外 = 追加書き込み 0　　= 上書き
RcvFileName	string	指定する受信ファイル名文字列のポインタ
FieldData	string	指定するフィールド情報文字列のポインタ 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） フィールド情報を指定しない場合は、フィールド情報文字列を保存する領域（256 ﾊﾞｲﾄ）を確保し、‘¥0’で初期化してその領域のポインタを設定します ＰＤ4 プログラム，ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
Int32  hCom, ret;
syting FieldData;

FieldData = new String( ‘¥0’, 255);    //255ﾊﾞｲﾄの領域確保
ret = BhtYmText.Receive( hCom, “”, 0, 0, “TANA.DAT”, FieldData );
if( ret = 0 )
{
  //受信終了処理
}
```

⑤ *Delphi 言語*

function RcvFileChk(hCom: Longint; FolderName: PAnsiChar; RcvMode, WrMode: Longint;
RcvFileName, FieldData: PAnsiChar): Longint;

引数	型	説明
hCom	Longint	通信ハンドル(OpenCom が返した値)
FolderName	PAnsiChar	受信ファイルを保存するフォルダ文字列のポインタ （NULL 文字を指定するとデフォルトフォルダになります）
RcvMode	Longint	0 以外 = 受信終了後次の受信待ちになります（連続受信） 0 = 受信終了で戻ります
WrMode	Longint	0 以外 = 追加書き込み 0 = 上書き
RcvFileName	PAnsiChar	指定する受信ファイル名文字列のポインタ
FieldData	PAnsiChar	指定するフィールド情報文字列のポインタ 「, 」を区切り記号にした文字列になります。（ 6,13,1 ） フィールド情報を指定しない場合は、フィールド情報文字列を保存する領域（256ﾊﾞｲﾄ）を確保し、 Chr(0)で初期化してその領域のポインタを設定します ＰＤ4 プログラム, ＦＮ３プログラムの場合は、NULL 文字になります。

戻り値	説明
0	**正常終了**
0 以外	**エラーコード表参照**

例）
```
var
    hCom, ret:   Longint;
    FieldData:   Array[0..255] of AnsiChar;  //256ﾊﾞｲﾄの領域確保

    FillChar(FieldData,SizeOf(FieldData),Chr(0));
    ret := BhtYmText.Receive( hCom,  '', 0, 0, 'ANA.DAT', FieldData );
    if ret = 0 then
    begin
      //受信終了処理
    end;
```

5-14 エラーコード表

Ｓｅｎｄ，Ｒｅｃｅｉｖｅ関数の戻り値のエラーコードです。

エラーコード	内　容
０	正常終了
１	取消
２	中断
３	指定通信ハンドル不良
４	実行エラー（通信中）
５	指定フォルダが存在しない
６	送信ファイルが無い
７	ﾌｨｰﾙﾄﾞﾌｧｲﾙ(ﾃﾞｰﾀ)ｴﾗｰ
８	ＢＨＴファイル名が不良
９	送信データエラー
１０	受信ファイル指定がない
１１	ディスク空き容量不足
１２	ファイルアクセスエラー
１３	タイムアウト
１４	リトライエラー
１５	受信フィールドエラー
１６	受信データエラー
１７	受信バッファ読み取りエラー
１８	送信バッファ書き込みエラー
１９	受信ファイルが違います
２０	指定フィールドと違います
９９	ライセンスエラー

６．ハンディターミナルへのファイル送信

データファイル名：　　＜半角１～８文字の英数字＞．＜拡張子（半角１～３文字の英数字）＞
ＣＳＶデータファイルを送信する場合は、拡張子ＣＳＶを使用します。
ＰＤ４，ＰＤ３，ＥＸ３，ＦＮ３，ＦＬＤ，ＥＸＥ，ＢＡＴ　の拡張子は使用できません。
拡張子を省略する場合はピリオドも取ってください。

Ｓｅｎｄ関数のFieldData引数にフィールド情報を指定しますが、NULL文字を指定した場合は、フィールドファイルを使用します。
フィールドファイル名：　　＜データファイル名と同じファイル名＞．ｆｌｄ
・フィールドファイルの書式
＜項目１桁数＞，＜項目２桁数＞，＜項目３桁数＞，　・・・　CRLF

ＴＡＮＡ．ＤＡＴというデータをハンディターミナルへ送信する場合、拡張子がｆｌｄのフィールドファイルＴＡＮＡ．ｆｌｄを作成する必要があります。フィールドファイルは、送信するデータ項目の桁数情報です。

例）商品コード１３桁と数量４桁のデータを送信する場合
Ｓｅｎｄ関数のFieldData引数：　“１３，４”
フィールドファイルデータ：　　１３，４CRLF

・固定長データ（ＴＡＮＡ.ＤＡＴ）
商品コード　　｜数量
４９０１７８０３３１２０８０００１
４９０１７８０３３１２０８００１０

・ＣＳＶデータ（ＴＡＮＡ．ＣＳＶ）
商品コード　　｜数量
“４９０１７８０３３１２０８”,”１”
“４９０１７８０３３１２０８”,”１０”

７．ハンディターミナルからの受信データファイル

ハンディターミナル内のファイル名に「**ＹＹＭＭＤＤ**」があると、**ＹＹＭＭＤＤの位置に日付をいれて受信データのファイル名にします。**

例）2011/01/01 に受信した場合
TAYYMMDD.TXT ⇒ TA110101.TXT
YYMMDD.CSV ⇒ 110101.CSV
YYMMDDAB.DAT ⇒ 110101AB.DAT

受信したデータは、固定長デキストファイルまたはカンマ区切りテキストファイルになります。
カンマ区切りテキストファイルになるのは、「カンマ区切りデータ受信」の設定または拡張子が「．ＣＳＶ」の場合です。

例）次のようなデータの場合
処理日：１０桁　棚 No：６桁　ＪＡＮコード：１３桁　数量：６桁（符号付）

Receive 関数で受信するフィールド情報は、「10 6 13 7」になります。

① 固定長ファイル

処理日　棚Ｎｏ　ＪＡＮコード　数量

２００６／０１／０１０００００１４９０１７８０３３１２０８＋０００００１
２００６／０１／０１０００００１４９０１７８０３３１２０８＋０００００１
・
・
・
２００６／０１／０１０００００１４９０２０１１６００２０９＋００００００１０

② 「カンマ区切りデータ受信」または拡張子が「.ＣＳＶ」の場合　（*5-8 SetComMode* 参照）

```
２００２／０４／０１，０００００１，４９０１７８０３３１２０８,＋０００００１
２００２／０４／０１，０００００１，４９０１７８０３３１２０８,＋０００００１
                        ・
                        ・
                        ・
２００２／０４／０１，０００００１，４９０２０１１６００２０９,＋０００００１
```

③ 「カンマ区切りデータ受信」または拡張子が「.ＣＳＶ」「カンマ区切りデータを引用符（“）で括る」を設定した場合（*5-8 SetComMode* 参照）

```
“２００２／０４／０１”，“０００００１”，”４９０１７８０３３１２０８”,”＋０００００１”
”２００２／０４／０１”，“０００００１”，”４９０１７８０３３１２０８”,”＋０００００１”
                        ・
                        ・
                        ・
”２００２／０４／０１”，“０００００１”，”４９０２０１１６００２０９”,”＋０００００１”
```

８．Ｅｘｃｅｌにおける使用例

フィールドの書式を文字列にしてWorkbooks.OpenTextメソッドを使用します。

```
Dim hWnd As Long, hCom As Long
Dim RcvFolder As String, RcvFileName As String, FieldData As String
Dim MyField(1 To 256)
Dim I As Integer, p As Integer

RcvFolder = String( 255, vbNullChar )  '255ﾊﾞｲﾄの領域を確保
RcvFileName = String( 15, vbNullChar ) '15ﾊﾞｲﾄの領域を確保
FieldData = String( 255, vbNullChar )  '255ﾊﾞｲﾄの領域を確保
hWnd = GetActiveWindow();
hCom = OpenCom( hWnd, "COM1", 115200 )
If hCom <> -1 then
    If Receive(hCom, "", 0, 0, RcvFileName, FieldData) = 0 Then
        RcvFileName = Left(RcvFileName, InStr(RcvFileName, Chr( 0 ) ) -  1 )  ‘文字数確定
        FieldData = Left(FieldData, InStr(FieldData, Chr( 0 ) ) -  1 )        ‘文字数確定
        If GetDefFolder( 0, RcvFolder ) <> 0 Then
            RcvFolder = Left( RcvFolder, InStr( RcvFolder, Chr( 0 ) ) -  1 )  ‘文字数確定
            p = InStr(1, RcvFileName, ".")
            If MidB(RcvFileName, p) = ".CSV" Then     ‘ CSV ファイルを Excel シートに読み込む
                 ‘ 全フィールドのデータ形式を文字列にして取り込む
                For i = 1 To 256
                    MyField(i) = Array(i, 2)
                Next i
                 ‘ Excel シートに読み込み
                Workbooks.OpenText Filename:= RcvFolder & RcvFileName, _
                                   DataType:=xlDelimited, Comma:=True, FieldInfo:=MyField()
            End If
        End If
    End If
    CloseCom(hCom)
End If
```

９．ＡＣＣＥＳＳにおける使用例

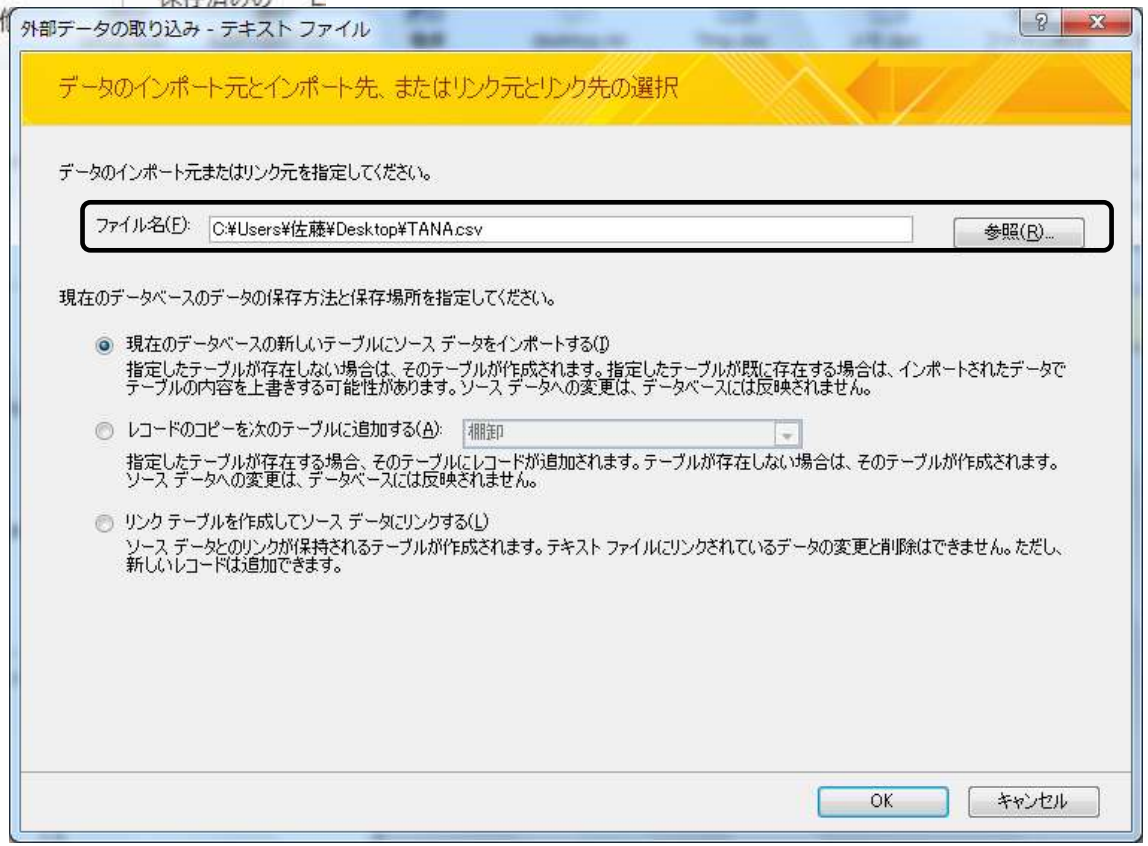
ｲﾝﾎﾟｰﾄ定義を作成して TransferText メソッドを使用します。

```
Dim hCom As Long
Dim RcvFolder As String, RcvFileName As String, FieldData As String

RcvFolder = String( 255, vbNullChar )  '255ﾊﾞｲﾄの領域確保
RcvFileName = String( 15, vbNullChar ) '15ﾊﾞｲﾄの領域確保
FieldData = String( 255, vbNullChar )  '255ﾊﾞｲﾄの領域確保
hCom = OpenCom( hWnd, "COM1", 115200 )
If hCom <> -1 then
    If Receive(hCom, "",0,0, RcvFileName, FieldData) = 0 then
        RcvFileName = Left(RcvFileName, InStr(RcvFileName, Chr( 0 ) ) -  1 )  ‘文字数確定
        FieldData = Left(FieldData, InStr(FieldData, Chr( 0 ) ) -  1 )        ‘文字数確定
        If GetDefFolder( 0, RcvFolder ) <> 0 Then
            RcvFolder = Left( RcvFolder, InStr( RcvFolder, Chr( 0 ) ) -  1 )  ‘文字数確定
            If RcvFileName = "TANA.CSV" Then      ‘ TANA.CSV ファイルをテーブルに読み込む
                DoCmd.TransferText acImportDelim, "TANA ｲﾝﾎﾟｰﾄ定義", "棚卸データ", RcvFolder & RcvFileName
            End If
        End If
    End If
    CloseCom(hCom)
End If
```

ｲﾝﾎﾟｰﾄ定義名（作成方法は後述）　テーブル名　受信テキストファイル名

＊Ａｃｃｅｓｓにおけるインポート定義の作成手順

② インポートするサンプルデータをＨＴで収集し、ＰＣに転送してテキストファイル用意します。
② メニューバーの［外部データ］をクリックして、［テキストファイル］をクエイックする。
③ インポートするファイルと保存場所を指定して ＯＫ ボタンをクリックします。



インポートするファイルとと保存場所をを指定して、ＯＫ ボタンをクリックします。

④ テキストインポートウィザードを表示しますので、 設定(V)_ ボタンをクリックしてください。
⑤ インポート定義画面を表示しますので、 保存(V)_ ボタンをクリックして、インポート定義名を入力して ＯＫ ボタンをクリックして保存してください。これでインポート定義ファイルが作成されました。