VBMan

Control for RS-232C

Version 4.00

TechKnowledge

VBMan Control for RS-232C

はじめに

VBMan Control for RS-232C version 4.00をお買いあげくださり誠にありがとうございます。当製品はMicrosoft Visual Studio.NET等COMコンポーネントをサポートする言語からご利用いただけるRS-232c通信をサポートするカスタム・コントロール製品です。以下は当コンポーネントの概要・特徴です。

- 信頼性と実績
  当コンポーネントは1994年の16bit/VBXコンポーネントの時代から32bitOCX,ActiveXコントロールの時代を経て、このたびVisual Studio.NETのCOMコンポーネント形式をサポートすることになりました。RS-232C用ソフトウェア・コンポーネントとして多くのお客様のアプリケーションに組み込まれており、多数の実績がございます。
- モジュール・サイズの極小化
  当コンポーネントはVisual Studio.NETでサポートされるATL version 9.0を使ってビルドされています。最新のATL技術を利用しコンポーネントの信頼性、配布の簡便性のためにモジュールサイズの極小化を実現しています。
- バイナリ・データ送信メソッドのサポート
  Win32環境では内部漢字コードとしてUnicodeが採用されており、文字列は2バイトで管理されるようになりました。従来の16bit環境ではバイナリ・データも文字列型で管理できましたが、Win32環境ではバイナリ・データは専用のデータ型を使うように仕様が変更になりました。この仕様変更に対応してVBMan Control for RS-232CではVisual Basic等のByte型を送受信できるメソッドをサポートしています。
- C#言語のサポート
  当バージョンからVisual Studio.NETの新しい言語であるC#をサポートします。サンプルコードを添付しました。
- Unicode文字列送受信のサポート
  当バージョンからUnicode形式のデータをRS-232cライン上にそのまま展開して送受信するメソッドをサポートします。
- 通信イベントのサポート
  当コントロールは通信イベント監視スレッドにより、回線状況、ファイル転送の完了、エラー・ステータス、送受信文字などを通信イベントとしてユーザー・プログラムに通知します。また、当バージョンから受信イベントのパラメータとしてVariant型を選択可能となりました。バイナリ受信時が必要な場合にはコードが簡便になります。

- デバック機能
  VcDebugTraceプロパティを設定すると送受信データをトレース・プログラムに16/10進表示することができます。アプリケーションのデバッグ効率を改善します。
- マルチ・スレッドによるバックグラウンドファイル転送
  Xmodem Check Sum, Xmodem CRCによるファイル転送が可能です。これらの処理はWIN32マルチスレッドにより効率よく実行されます。
- スレッド優先順位の設定
  さまざまな動作環境に対応するために通信管理スレッド、ファイル転送スレッドの優先順位をプロパティで設定することが可能です。そう
- 送受信タイム・アウトの設定
  通常は感知することのできない接続された通信機器の電源断などでも送受信タイム・アウトをプロパティで指定することにより、アプリケーションのハング・アップ等を回避できます。

開発ライセンス

開発ライセンスとは、本製品1ライセンスを1台のパーソナル・コンピュータ・システムで1開発者が利用することが出来る権利です。当製品のディスケットを複製して、複数人でのご使用等は著作権法違反となり罰せられます。ご注意ください。

- 当ソフトウェア製品は、著作権法及び国際著作権条約をはじめ、その他の無体財産権に関する法律ならびに条約によって保護されています。
- 当ソフトウェアに対するリバースエンジニアリング及び、改変は一切禁止します。
- 本製品をラップする形で作成した同様の機能を持ったカスタム・コントロール製品の販売は禁止いたします。
- 当製品の使用権はいかなる方法によっても第三者に譲渡および貸与することは出来ません。
- 使用権は当製品を開梱したときに発効します。商品パッケージ開梱後の返品はできませんので予めご了承ください。
- 使用権は以下のいずれかの事由が起こった場合に消滅します。
  購入者が製品に同封されているユーザー登録書を返送しない場合。
  購入者が使用規定に違反した場合。
  プログラム・ディスク、印刷物などを使用権の範囲外の目的で複製した場合。

ランタイムライセンス

ランタイムライセンスとは弊社製品のコンポーネントをお客様のアプリケーションと一緒に配布して動作させる使用権です。当製品は1開発ライセンス＋1ランタイムライセンスのパッケージとさせていただいております。追加ランタイムライセンスにつきましてはシステム・ラボまでお問合わせください。

保証規定

当製品、および付随する著作物に対して商品性及び特定の目的への適合性などについての保証を含むいかなる保証もそれを明記するしないに関わらず提供されることはありません。

当製品の著作者及び、製造、配布に関わるいかなる者も、当ソフトウェアの不具合によって発生する損害に対する責任は、それが直接的であるか間接的であるか、必然的であるか偶発的であるかに関わらず、負わないものとします。それは、その損害の可能性について、開発会社に事前に知らされていた場合でも同様です。

ユーザー・サポート

- ユーザー登録

  この製品には、ユーザー登録用紙を添付しています。お買い上げのあと、できるだけ早い機会に必要事項をご記入の上、販売会社システム・ラボまでファックスでご送付ください。このユーザー登録が行われていないと、ユーザーサポートが受けられない場合があります。必ずご返送をお願いいたします。

- お問い合わせの方法

  どうしても解決できない問題が発生した場合には、システム・ラボの技術サポートをご利用ください。あらかじめパッケージに添付される調査依頼書にお問い合わせ事項を記入していただき、それをファックス、インターネット・メールまたはお手紙でお送りいただければ、折り返しご連絡をさせていただきます。**当製品につきましては、製品の性格上複雑なやりとりになる場合が多いため、電話によるユーザーサポートはいたしておりませんので、ご了承をお願いいたします。**また、問い合わせの内容によっては、調査などのために回答に時間がかかる場合がありますので、かさねてご了承をお願いいたします。

- 登録内容の変更について

  転居などによるご住所や電話番号など、登録内容に変更が生じた場合には、郵送またはファックスにて、販売会社システム・ラボまでご連絡をいただきますようお願いいたします。なお、電話による口頭での連絡変更は受けかねますので、よろしくお願いいたします。

- 併用される他社製品について

  当社製品と併用される、他社製品の使用方等についてのご質問をお受けすることがあります。しかし、他社製品に関しましてはお答えできない場合があります。他社製品につきましては、該当開発・販売会社にご連絡ください。

- 無償サポート期間について

  無償サポート期間はユーザー登録後、最初のお問い合わせから90日間となっております。有償サポートにつきましては販社システム・ラボにて承っておりますのでご連絡ください。

- サポートのパフォーマンスについて

  簡単なお問い合わせであれば1労働日以内を目標にサポートをいたします。お問い合わせの内容により調査などのために回答に時間がかかる場合がありますので、あらかじめご了承ください。また弊社が別途定めた定休日にはサポートも休止する場合があります。サポートに優先順位は

ありません。当着順に処理いたしますのでよろしくお願いいたします。

- 登録内容の変更について
  転居などによるご住所や電話番号など、登録内容に変更が生じた場合には、郵送またはファックスにて、販売会社システム・ラボまでご連絡をいただきますようお願いいたします。なお、電話による口頭での連絡変更は受けかねますので、よろしくお願いいたします。

販売元

株式会社システムラボ

東京都北区田端 6-1-1　田端アスカタワー　12階

電話:	03-5809-0839
ファックス:	03-4587-9261
サーポートメール:	support@systemlab.co.jp
Web:	www.systemlab.co.jp

開発元/サポート

（株）テクナレッジ

東京都世田谷区駒沢2丁目16番1号　サンドービル9F

電話:	03-3421-7621
ファックス:	03-3421-6691
サポートメール:	support@techknowledge.co.jp
Web:	www.techknowledge.co.jp

インストール

VBMan Control for RS-232Cのインストールについて説明します。

システム条件

VBMan Control for RS-232C 以下のオペレーティング・システムが動作するパーソナル・コンピュータ環境で動作します。各オペレーティング・システムにはサービス・パックが配布されておりますのでそちらのインストールもご検討ください。また、今後の動作環境につきましては弊社 web サイトにてご確認ください。

- Windows 95/98/98 Second Edition/Me
- Windows NT バージョン 4.0
- Windows 2000
- Windows XP
- Windows 7/8/8.1/10

VBMan Control for RS-232C は以下のコントロール・コンテナ・アプリケーションでの動作をサポートしています。

- Microsoft Visual Basic.NET
- Microsoft Visual C#
- Microsoft Visual Basic 5.0-6.0
- Borland Delphi 3.x/4.0/5.0/6.0
- Microsoft Access 2010-2016
- Microsoft Visual C++ 7.0/6.0/5.0
- Microsoft Excel 2010-2016

Microsoft Excel からの利用方法は弊社ウェブに別ドキュメントをご用意してありますのでご参照ください。

http://www.techknowledge.co.jp/pdf/how_to_serial_on_excel.pdf

セットアップの起動

VBMan Control for RS-232C の CD-ROM をドライブに挿入しセットアップ・プログラム(setup.exe)を起動します。セットアップ・プログラムではインストール・ディレクトリを指定するとコントロール・パックのインストール完了します。フォルダーVBMan Control for RS-232C が作成され、ヘルプ、サンプル等がメニューに登録されます。またインストール実行時には管理者権限が必要です。

インストール・ファイル一覧

以下に当製品のインストールされるファイルの一覧を示します。システム・ディレクトリは Windows95 のデフォルトで表示します。ドットから始まるディレクトリ表示は VBMan のインスト

ール・ディレクトリになります。32BIT Windows デフォルトのインストール・ディレクトリは c:¥Program Files¥TechKnowledge¥VBMan Controls for RS-232C 4.0 となります。

モジュールの再配布の欄にご注意ください。再配布不可と記載されるモジュールはランタイムライセンスを取得された場合に配布可能となるファイルです。再配布不可と記載されるファイルにつきましては開発環境のみでのご利用可能となります。再配布件なしに配布した場合ソフトウェアの著作権を侵害していることになり日本国憲法で罰せられる場合があります。

モジュール名	概　　要	再配布
¥WINDOWS¥SYSTEM¥VBMCOM32.OCX	RS-232C 通信 ActiveX コントロールモジュール	可
¥WINDOWS¥SYSTEM¥VBMSHM32.DLL	共有メモリーモジュール	可
.¥VBMTRACE.EXE	デバッグ・トレース表示ユーティリティ	不可
.¥VBMCOM32.HLP	オンラインヘルプ	不可
.¥VBMCOM32.TXT	注意書	不可
.¥SAMPLES¥ACCESS¥BARCODE.MDB	Access 用サンプル	不可
.¥SAMPLES¥DELPHI¥BARCODE.*	Delphi 用バーコード・サンプル	不可
.¥SAMPLES¥DELPHI¥BCSAMP.*	Delphi 用バーコード・サンプル	不可
.¥SAMPLES¥DELPHI¥BINARY.*	Delphi 用バイナリ送受信サンプル	不可
.¥SAMPLES¥DELPHI¥BINSAMP.*	Delph 用バイナリ送受信サンプル	不可
.¥SAMPLES¥IE¥BARCODE.HTM	インターネット・エクスプローラー用サンプル	不可
.¥SAMPLES¥VB6¥BARCODE.VBP	VB6.0 用ターミナル･サンプル・フォーム	不可
.¥SAMPLES¥VB6¥BARCODE.FRM	VB 6.0 用バーコード読み取りサンプル・フォーム	不可
.¥SAMPLES¥VB6¥TERM.VBP	VB6.0 用ターミナル・サンプル・プロジェクトファイル	不可
.¥SAMPLES¥VB6¥TERM.FRM	VB6.0 用ターミナル･サンプル・フォーム	不可
.¥SAMPLES¥VB6¥BARCODE.VBP	VB6.0 用バーコード読み取りサンプル・プロジェクトファイル	不可
.¥SAMPLES¥VB6¥BARCODE.FRM	VB 6.0 用バーコード読み取りサンプル・フォーム	不可
.¥Samples¥VB.NET¥TERM¥*.*	VB.NET 用ターミナルサンプル	不可
.¥Samples¥CS¥Term¥*.*	C#用ターミナルサンプル	不可
.¥Samples¥CS¥Binary¥*.*	C#用バイナリ送受信サンプル	不可

プロジェクト開始

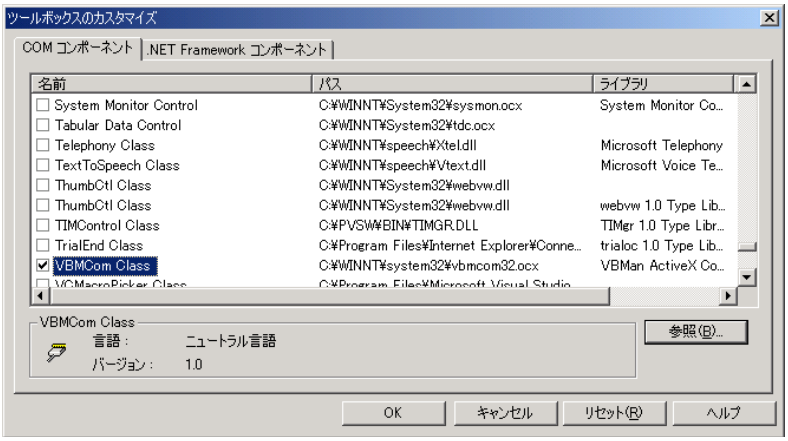
ActiveX コントロールは多くのコンテナ・アプリケーションから利用可能ですが代表的なコンテナとして Visual Basic.NET, Visual Basic 6.0, Borland Delphi での利用方法を説明します。

Visual Basic.NET

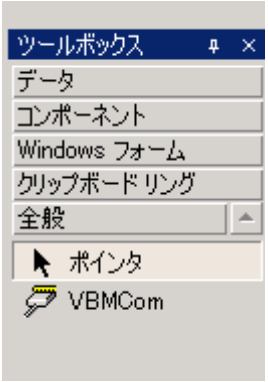
Microsoft Visual Basic.NET から VBMan Control for RS-232C を使う方法を説明します。

① Visual Studio.NET を起動します。
② 新規プロジェクトをファイルメニューから選択します。
③ 言語を Visual Basic.NET を選択します。
④ プロジェクトタイプは「Windows アプリケーション」を選択します。
⑤ ツールボックスから VBMan を追加したいタブを選択するか、新規タブを作成し、該当タブを開きます。
⑥ マウスの右クリックで表示されるメニューで「ツールボックスのカスタマイズ」を選択します。
⑦ 「COM コンポーネント」タブを選択します。(デフォルト)
⑧ 「参照」ボタンを押し、システム・ディレクトリにある vbmcom32.ocx を選択します
⑨ 追加された「VBMCom」コントロールを Windows フォームにドラッグして利用します。

以下は手順⑦の実行画面例です。



以下は全般タブに VBMan Control for RS-232C を追加した例です。



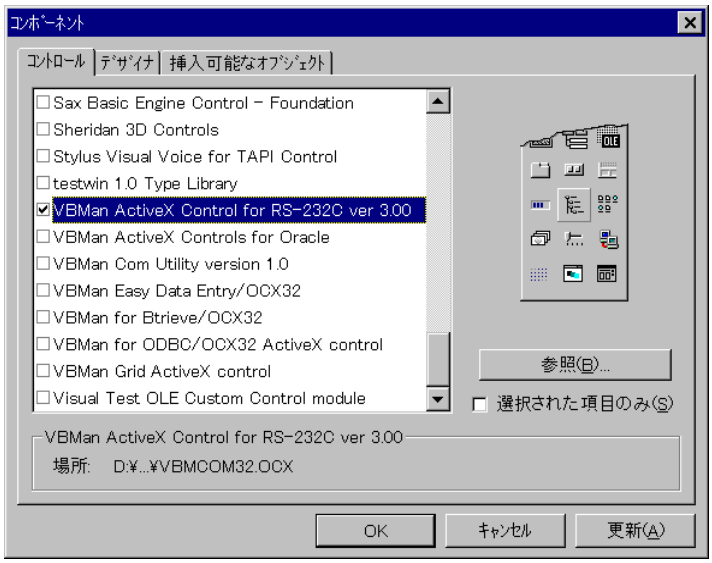
Visual C#

Visual C#の場合は手順③で選択する言語を Visual C#とする以外は Visual Basic.NET の場合と同一です。

Visual Basic 5.0/6.0

Microsoft Visual Basic 5.0/6.0 から VBMan Control for RS-232C を使う方法を説明します。

① Visual Basic を起動します。
② VBMan Control for RS-232C を組み込みたいプロジェクトをオープンします。
③ 「プロジェクト」メニューから「コンポーネント」を選択します。
④ リスト・ボックスから「VBMan Control for RS-232C ver 4.00」を選択し、OK ボタンを押します。[1]



⑤ Visual Basic のツール・ボックスに VBMan Control for RS-232C コントロールのアイコンが追加されます。

---

[1] リスト・ボックスにこの項目がない場合は、インストールに失敗していると思われます。当マニュアルのトラブルシュートを参照して環境を整えて再インストールをしてください。

⑥ 通信をしたいフォームに VBMan Control for RS-232C コントロールをマウス・ドラッグで配置します。実行時には不可視になりますのでフォームのデザイン時に邪魔にならない適当な場所に置いてください。

⑦ マウスの右ボタンをクリックして「プロパティ」を選択します。以下のようなダイアログが表示されますので、通信条件など基本的な設定をします。

⑧ 通信条件の設定が終わったら Visual Basic のアプリケーションが通信を開始したいイベントに通信のメソッドを記述することでアプリケーションを完成します。

Borland Delphi

Borland Delphi から VBMan Control for RS-232C を使用する方法を説明します。

① メニューから「コンポーネント」を選択し、「インストール」を選択します。
② 「コンポーネントのインストール」ダイアログで「ActiveX」ボタンを選択します。
③ 「登録されているコントロール」のリスト・ボックスから「VBMComm Class 」を選択して「OK」ボタンを押します。
④ ツール・バーの ActiveX タブに VBMan Control for RS-232C のアイコンが登録されていることを確認します。
⑤ アプリケーションで通信をしたいイベント等に通信メソッドを呼び出してアプリケーションを完成します。

ご注意

VBMan for RS-232C/OCX32 version 2.x では Borland Delphi 2.0J で UniCode の問題からバイナリ通信ができないという問題があり、生成される Object Pascal のラッパー・クラスを修正して対応していましたが、この方法は煩雑なので当バージョンからは SendBinaryBytes, RecvBinaryBytes メソッドを使ってバイナリ・データを送受信する方法をサポートすることに変更しました。Delphi 3.x 用のバイナリ送受信サンプルが添付されていますのでご参照ください。

コンパチビリティ

ActiveX 版(version 3.X)からの移行

コンポーネントの GUID は version 4.00 と version 3.X では同一の値を使っていますので、レイトバインディングで COM 呼び出しを解決する言語をお使いの場合は特に移行処理は必要ありません。

VB6 でコンポーネントをフォームに貼り付けて使っている場合追加したプロパティのコンパチビリティにより、フォームのロードに失敗する場合があります。そのような場合には単に新しいコンポーネントに再度張り替えていただくことで利用可能になります。

また、ラッパークラスを作成する言語をお使いの場合(Borland Delphi, Microsoft Visual C++など)、新しいメソッドおよびプロパティやイベントをご利用になる場合にのみ、再度ラッパーを生成させることが必要になる場合があります。ラッパークラスの再度作成方法はご利用の言語マニュアルをご参照ください。新規プロジェクトから当コントロールを設定して新たにラッパークラスを作成し、ラッパークラスを既存プロジェクトのものと入れ替える方法は有効と思われます。

OCX32 版からの移行

ここでは VBMan for RS-232C/OCX32 バージョン 2.x で作成された Visual Basic アプリケーションを VBMan Controls for RS-232C バージョン 3.0 へ移行する方法について説明します。

① VBP ファイルの変更
Visual Basic のプロジェクト・ファイルを notepad などのテキスト・エディターで開きます。
以下の Object=で始まる行を編集します。

編集前

Object={**EC436243-21B3-101C-A375-02350B10C007**}#1.0#0; VBMCOM32.OCX

編集後

Object={**25C0ABE6-C3C2-11D0-8FEC-0000E8A145B3**}#1.0#0; VBMCOM32.OCX

② フォーム・ファイルの編集
Visual Basic のフォーム・ファイルのコントロール名を編集します。

編集前

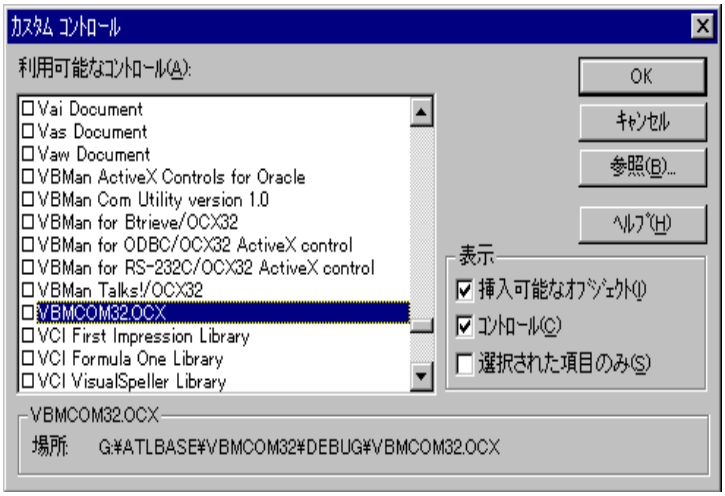
Begin VbmcommLib.Vbmcomm Vbmcomm1

編集後

Begin VBMCOM32LibCtl.VBMCom Vbmcomm1

以上の方法以外では Visual Basic でプロジェクトを読み込む時にエラーを無視してプロジェクト・ロードをして VBMan カスタム・コントロールをはり直す方法でも可能です。最初のプロジェクトをロードする前にプロパティの設定とコントロールの Name プロパティの値を調べておくと簡単にプロジェクトを移行できます。

マイクロソフトはカスタム・コントロールの CLSID を同じにして Visual Basic では自動的にコントロールをバージョン・アップする手法を提供していますが、当コントロールは別々の CLSID を割り振ってプロジェクトとソースを変更する方法をとっています。この方法の利点は旧バージョン 2.1 の VBMCOM32.OCX を別ディレクトリに待避した場合でもレジストリがおこなわれていれば、そちらも利用できるというメリットを優先したためです。以下は Visual Basic のカスタム・コントロール・ダイアログに新旧バージョンが登録されている状態のサンプルです。Visual Basic ではカスタム・コントロール名が単に VBMCOM32.OCX となっていることに注意してください。



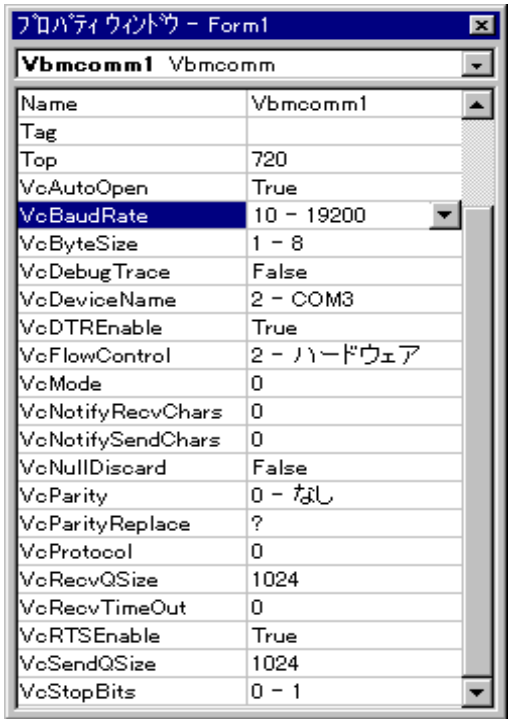
カスタム・プロパティ

VBMan Control for RS-232C では以下のカスタム・プロパティを設定することで通信アプリケーション通信条件やカスタム・コントロールの動作などを設定します。カスタム・プロパティで通信条件を設定することでコード記述量を減らすことができます。

カスタム・プロパティ名	詳　細
VcAutoOpen	通信の自動開始
VcBaudRate	通信速度
VcByteSize	通信データサイズ
VcDebugTrace	デバッグ・トレースの指定
VcDeviceName	通信ポート指定、(6ポート)
VcDTREnable	通信開始時にDTRラインをイネーブルにする。
VcFileTransferPriority	ファイル転送スレッドのプライオリティを指定
VcFlowControl	フロー制御指定
VcInBufferCount	受信バッファにある文字バイト数。
VcMode	テキスト・バイナリ指定
VcNotifyRecvChar	CommRecvイベントを発生させる受信文字バイト数
VcNotifySendComplete	CommSendイベントを発生させる
VcNullDiscard	ヌルを受信したら無視する
VcParity	パリティ
VcParityReplace	パリティ・エラー発生時に置換する文字を設定
VcProgress	ファイル転送の進捗を得る
VcProtocol	ファイル転送プロトコルを指定
VcRecvQSize	受信キュー・サイズ
VcRecvTimeOut	受信タイムアウト
VcRTSEnable	通信開始時にRTSをラインをイネーブルにする
VcSendQSize	送信キュー・サイズ
VcSendTimeOut	送信タイムアウト
VcShowErrorMessage	ファイル転送中のエラー・メッセージの表示方法を指定

VcStopBits	ストップ・ビット

以下は Visual Basic プロパティ・ウィンドウの例です。



VcAutoOpen

通信の開始にはポートのオープン・クローズ処理が必要ですが、このプロパティを True にした場合はそれらの処理をコントロールの生成、破壊と同時に自動的に処理します。このプロパティを False とした場合はアプリケーションに OpenComm,CloseComm メソッドを呼び出すコードが必要です。プロパティ・ウィンドウではチェック・ボックスをチェックした場合に AutoOpen モードとなります。

VcBaudRate

シリアル非同期通信の速度を設定します。ウィンドウズでサポートされる通信速度がプロパティ・ウインドウで選択できます。以下の通信速度をプロパティに設定します。プロパティのデータ型は short 型で列挙されます。以下の値が設定可能値です。

プロパティ値	ボーレート(bps)
0	75
1	110
2	150
3	300
4	600
5	1200
6	2400
7	4800
8	9600
9	14400
10	19200
11	28800
12	38400
13	57600
14	115200

VBMan Control for RS-232C はウィンドウズの API を呼び出して通信します。基本的に Windows の API は機種依存はないことが前提になっていますが、NEC の 98 シリーズでは機種によって最高通信速度は異なります。 一部の機種では 19200bps の速度はサポートできない、などの制限があります。

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcByteSize

7ビットまたは8ビットを指定します。プロパティのデータ型は short 型です。列挙型のプロパティで以下の値を設定します。

プロパティ値	バイト・サイズ
0	7bit
1	8bit

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcCustomBaudRate

通信速度は通常 VcBaudRate で固定値を設定しますが特定の機器に接続するような場合にはこちらのプロパティで任意の通信速度を設定して通信することが可能です。通信速度の上限・下限はご利用になるパソコンのシリアル通信チップや通信ボードの仕様により異なります。当プロパティ設定が 0 の場合は VcBaudRate プロパティを参照して通信速度を決定します。

VcDeviceName

通信ポートを選択します。プロパティのデータ型は列挙型で以下の値を設定可能です。[2]

プロパティ値	通信ポート
0	COM1
1	COM2
2	COM3
3	COM4
4	COM5
5	COM6
6	COM7
7	COM8
8	COM9
9	COM10

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcDebugTrace

プログラム開発時に有効なトレースの出力を指定します。このプロパティを True にするとデバッグ・トレースを出力します。デバッグ・トレースは Send/Recv メソッドについて送受信されたデータを vbmtrace.exe で 16 進または 10 進ダンプ表示します。デバッグ・トレースを使うとデータを vbmtrace.exe に転送するオーバー・ヘッドが発生します。タイミング・クリティカルなアプリケーションではこのメソッドを True に設定するとアプリケーションの動作が変わってしまう場合もありますのでご注意ください。

---

[2] 10 ポートまでプロパティ・ウィンドウで設定可能です。さらに大きな値を設定する場合はポートをオープンする前にコードで設定することで可能です。

VcDTREnable

DTR ラインの制御を指定します。プロパティを True に設定すると、通信開始時に DTR ラインをイネーブル状態に設定します。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。以下はサンプル・コードです。

```
Comm.VcDTREnable = True
```

VcFileTransferPriority

VBMan Control for RS-2323C ではファイル転送はバックグラウンドのスレッドで処理されます。このプロパティではファイル転送スレッドの優先順位を指定します。たとえば、ファイル転送中に同時に実行しているデータ・ベース・アクセスが極端に遅くなるような場合はこのプロパティの値を調節して CPU パワーの配分を調節することができます。以下の5段階の値をこのプロパティに設定することができます。このプロパティはファイル転送中に設定した場合は、実行中のファイル転送のプライオリティには有効ではありません。次回のファイル転送から設定した値が有効になります。

プロパティ値	意味
0	Lowest
1	Below Normal
2	Normal
3	Above Normal
4	Highest

VcFlowControl

フロー制御を設定します。なし、ハードウェア、XOn/XOff が設定可能です。通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。プロパティのデータ型は列挙型で以下の値を設定します。

プロパティ値	フロー制御
0	なし
1	XON/XOFF
2	ハードウェア

Visual Basic サンプル

```
Dim rc As Integer

Comm.VcDeviceName   = 1     ‘ COM2
Comm.VcFlowControl = 1     ‘ Xon/Xoff
rc = Comm.OpenComm
```

ハードウェアフロー制御を選択した場合、 Windows API レベルで RTS(request-to-send),DTR(data-terminal-ready)フロー制御を設定し、CTS(clear-to-send)タイムアウト、DSR(data-set-ready)タイムアウトは 30ms に設定します。

VcInBufferCount

プロパティを参照した時点での受信バッファ内にある受信データのバイト数を返します。

Visual Basic サンプル

```
Dim tmp$,txt$

While Comm.VcInBufferCount <> 0
  tmp$ = Comm.RecvString(1) ‘ 1文字ずつ受信して EOT を調べる。
  If tmp$ = EOT Then
    Text1.Text = txt$
    txt$ = “”
  Else
    txt$ = txt$ & Tmp$
  End If
Wend
```

VcMode

通信モードを設定します。整数型のプロパティで、以下の値を設定します。

プロパティ値	モード
0	バイナリ
1	テキスト

テキスト・モードの場合は行末に CR+LF(chr$(13) + chr$(10))を追加してデータを送信し、受信の時は CR+LF を削除してユーザー・アプリケーションに返します。バイナリ・モードでは、送受信データがそのまま転送されます。プロパティ・ウィンドウではチェック・ボックスを選択した状態がテキスト・モードとなります。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcNotifyRecvChars

通信バッファに受信したバイト数がこのプロパティに指定する値以上になると、CommRecv イベントが発生します。このプロパティに-1 を設定した場合、CommRecv イベントは発生しません。プロパティに 0 を指定した場合は受信したデータのバイト数に関係なく受信データが存在すれば CommRecv イベントが発生します。以下はサンプル･コードです。

Visual Basic サンプル

```
Comm.VcNoticyRecvChars = 1
```

```
...
...
‘ 一文字受信したら以下のイベントが発生する。
Sub Comm_CommRecv( RecvChar As String )
  If RecvChar = Chr$(13) Then
    ‘ 改行を処理
  End If
End Sub
```

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcNotifySendComplete

送信完了イベントの発生の有無を指定します。このプロパティを True に設定した場合、データの送信が完了して送信バッファが空になると CommSend イベントが発生します。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcNullDiscard

ヌル文字の処理方法を設定します。このプロパティを True に設定するとヌル文字を受信した場合は無視され、ユーザー・プログラムには返されなくなります。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcParity

パリティ・ビットを設定します。なし、偶数、奇数が選択可能です。プロパティのデータ型は列挙型です。以下の値が設定可能です。

プロパティ値	パリティ
0	なし
1	奇数
2	偶数

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcParityReplace

パリティ・エラーが発生した場合に置き換える文字を指定します。デフォルトは”?”です。通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcProgress

ファイル転送の進み具合を示します。プロパティのデータ型は short です。Xmodem プロトコルによる転送をしている場合は 128 バイト単位のパケット数がこのプロパティに設定されます。

VcProtocol

ファイル転送プロトコルを設定します。プロパティのデータ型は short です。以下の設定が可能です。Xmodem プロトコルを利用する場合、通信パラメータは 8bit,パリティ 1,フロー制御なしに設定する必要があることにご注意ください。

プロパティ値	プロトコル
0	Xmodem CRC, CheckSum,1K を自動切り替え
1	Xmodem CheckSum
2	Xmodem CRC
3	Xmodem 1K[3]

VcRecvEventType

受信イベントのタイプを選択します。

プロパティ値	意味
AsString	RecvComm イベントを発生させます。
AsVariantString	RecvCommVar イベントを発生させます。パラメータは Variant 型で文字列型データが含まれます。
AsVarinatBinary	RecvCommVar イベントを発生させます。パラメータは Variant 型ですが System.Byte 型データが含まれます。

Visual Basic 6.0 等イベントのパラメータとして Variant 型がサポートされていない言語では当プロパティの値を AsString としてご利用ください。Visual Basic.NET/Viasul C#におきましても Binary の配列はパラメータ型としてサポートされておりませんでしたので（当社調査）、VcNotrifyRecvChar に1以上の値を設定した場合にも1文字ずイベントが発生しますのでご注意ください。

VcRecvNotifyOnly

[3] Xmodem-1K を使用する場合、プロトコルではデータ・パケットは 1,024 バイトですが、送受信データがバッファリングされることを考慮して、VcSendQSize,VcRecvQSize プロパティを 1,200 バイト程度に設定してください。Xmodem-1K では送信側で STX を送信することでプロトコルを切り替えるため、受信側の Xmodem-1K の設定は必須ではありません。

このプロパティが False に設定された場合は、VcNotfyRecvChars プロパティで指定したバイト数が OnComm イベントの文字型パラメータでアプリケーションに返されます。
このプロパティが True に設定された場合は OnComm イベントのパラメータは常にヌル文字列になります。OnComm イベントでは必要なメソッドを呼び出して、受信キューにあるデータを読み出すことが可能になります。以下は OnComm イベントでバイナリ・データを受信するサンプルです。

Visual Basic サンプル

```
Private Sub VBMCom2_CommRecv(strRecv As String)
Dim b(0 To 2) As Byte, rc As Integer
rc = VBMCom2.RecvBinaryBytes(b)
Debug.Print b(0), b(1), b(2)
End Sub
```

VcRecvQSize

受信キュー・サイズをバイト単位で整数で指定します。通信開始前に設定される必要があります。デフォルトは 1024 バイトです。コードで設定する場合は以下のようになります。受信・キューの最大サイズはオペレーティング・システムで用意される通信デバイス・ドライバーの仕様に依存します。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

Visual Basic サンプル

```
Dim rc As Integer
Comm.VcRecvQSize = 128
Comm.VcSendQSize = 128
rc = Comm.OpenComm
If rc <> 0 Then
   MsgBox “通信オープンエラー” & Cstr(rc)
End If
```

VcRecvTimeOut

RecvString メソッドで文字列を受信するとき、受信バイト数をパラメータで指定した場合に受信タイムアウトをこのプロパティで指定可能です。単位はミリ・セカンド(1/1000 秒)です。プロパティのデータ型は Long 型です。タイム・アウトはエラー・イベントに ERR_RECV_TIMEOUT が発生し、RecvString にはその時点まで受信した文字列が返されます。４９日以上連続稼動しているパソコンでは動作しない可能性もありますので、ご注意ください。[4]

Visual Basic サンプル

[4] タイム・アウトの計測に Win32API の GetThickCount を使っています。この API は 49 日でカウントがラップするので、連続稼動はできない API です。現実にはタイムアウトで 49 日以上を指定することは希と思いますが念のため記述しています。

```
Dim r$

VBManCom1. VcRecvTimeOut = 1000      ‘一秒
r$ = VBManCom1.RecvString(10)
```

VcRTSEnable

このプロパティを True に設定すると通信開始時に RTS ラインをイネーブルにします。通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。以下はサンプル･コードです。

```
Comm.VcRTSEnable = True
```

VcSendQSize

送信キューのサイズをバイト単位で整数で指定します。デフォルトは 1024 バイトです。このプロパティは通信開始前に設定される必要があります。コードで設定するサンプルは、VcRecvQSize を参照してください。また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcSendTimeOut

送信タイムアウトを msec 単位で指定します。VBMan が通信のために呼び出す API のパフォーマンス（COMM デバイス・ドライバーも関係します）によっては正確な msec 単位でのタイムアウトができない場合もありますので、あらかじめご了承ください。プロパティの値に0を指定した場合はタイムアウトはしないで送信が完了するまで待ちになります。

Visual Basic サンプル

```
Dim rc As Integer
Const ERR_SEND_TIMEOUT = 158

Comm.VcSendTimeOut = 1000  ‘タイムアウトを一秒
rc = Comm.SendString(“atz” & Chr(13) & Chr(10) )
If rc = ERR_SEND_TIMEOUT Then
   Msgbox “送信タイムアウトです”
End If
```

VcShowErrorMessage

ファイル転送中のエラーが発生した場合のメッセージの表示方法を指定します。このプロパティに True 値を設定した場合にはエラー・メッセージをメッセージ・ボックスで表示します。False 値を指定した場合にはメッセージ・ボックスは表示されません。この場合、CommTransferEnd イベントに通知されるエラー・ステータスでファイル転送の完了を知ることができます。

VcStopBits

ストップ・ビットを設定します。1,1.5,2 ビットを設定可能です。プロパティのデータ型は short です。設定は以下の対応になります。

プロパティ値	ストップビット
0	1
1	1.5
2	2

また、通信ポートをオープンしている状態でこのプロパティをセットした場合、変更は有効になりません。次回のポート・オープン時に変更が反映されます。

VcWatchPriority

VBMan Control for RS-2323C では通信イベントの監視をバックグラウンドのスレッドで処理しています。このプロパティでは通信イベント監視スレッドの優先順位を指定します。たとえば、通信中に同時に実行しているデータ・ベース・アクセスが極端に遅くなるような場合はこのプロパティの値を調節して CPU パワーの配分を調節することができます。以下の5段階の値をこのプロパティに設定することができます。
このプロパティはポートのオープン時にスレッドが起動されるときに参照されます。ポート・オープン中にこのプロパティを設定した場合には設定値は有効にはなりません。ポートをオープンする前にこのプロパティの値を設定してください。

プロパティ値	意味
0	Lowest
1	Below Normal
2	Normal
3	Above Normal
4	Highest

カスタム・メソッド

ここでは VBMan Control for RS-232C で利用可能なメソッドについて説明します。Visual Basic 等からこれらのカスタム・メソッドの呼び出しコードを記述することにより、通信アプリケーションを作成します。

メソッド名	詳　細
AbortTransfer	ファイル転送を中断
ClearBreak	ブレーク状態のクリア
ClearDTR	DTRラインのクリア
ClearRTS	RTSラインのクリア
CloseComm	通信の終了
FlushComm	通信キューのデータを破棄
GetCTS	CTSライン状態を得る
GetDSR	DSRライン状態を得る
GetRLSD	RLSDの状態を得る
GetRing	Ringの状態を得る
OpenComm	通信の開始
RecvBinaryBytes	Byte配列を送信
RecvFile	ファイル受信の開始
RecvString	文字列を受信
RecvUnicodeString	Unicode文字列を受信します。
SendBinaryBytes	バイナリを含むByte配列を送信
SendBreak	プレーク信号送信
SendByte	現在の送信バッファにあるデータを送信
SendChar	１文字を送信
SendFile	ファイル送信の開始
SendString	文字列を送信
SendUnicodeString	Unicode文字列としてデータをラインに出力します。
SetDTR	DTRラインをオンにする
SetRTS	RTSラインをオンにする

AbortTransfer

書式
AbortTransfer () As Integer

戻り値
メジャー通信エラー･コード。エラー･コード一覧を参照してください。

解説
バイナリ･ファイル転送を中断します。接続するソフト、タイミングによって切断の結果が異なることがありますので、ご注意ください。

Visual Basic サンプル
```
Dim rc As Integer
rc = VBManComm1.AbortTransfer
```

ClearBreak

書式
ClearBreak() As Integer

戻り値
メジャー通信エラー･コード。エラー･コード一覧を参照してください。

解説
通信ポートをブレーク状態から通常の通信状態にもどします。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

Visual Basic サンプル
```
Dim rc As Integer
rc = VBManComm1.ClearBreak
```

ClearDTR

書式
ClearDTR() As Integer

戻り値
メジャー通信エラー･コード。エラー･コード一覧を参照してください。

解説
DTR ラインをオフにします。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

Visual Basic サンプル
```
Dim rc As Integer
```

rc = VBManComm1.ClearDTR

ClearRTS

書式
ClearRTS() As Integer

戻り値
メジャー通信エラー･コード。エラー･コード一覧を参照してください。

解説
RTS ラインをオフにします。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

Visual Basic サンプル
```
Dim rc As Integer
rc = VBManComm1.ClearDTR
```

CloseComm

書式
CloseComm() As Integer

戻り値
メジャー通信エラー･コード。エラー･コード一覧を参照してください。

解説
通信ポートをクローズします。VcAutoOpen プロパティが True の場合は使用できません。以下はサンプルです。戻り値はエラー・イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

Visual Basic サンプル
```
Dim rc As Integer
rc = VBManComm1.CloseComm
```

FlushComm

通信キュー・バッファをフラッシュします。

書式
FlushComm(*QueueType* As Integer) As Integer

パラメータ
メソッドの動作を指定します。以下の値が指定可能です。

値	キュー・タイプ

1	実行中の送信処理を中断します
2	実行中の受信処理を中断します
4	送信キューをクリアします
8	受信キューをクリアします

戻り値
MajorErrorCode の値を返します。

解説
通信キュー・バッファのデータを破棄します。送受信両方のキューを 1 度のメソッド呼び出しで破棄する場合は値12を指定します。

GetCTS

書式
GetCTS() As Integer

戻り値
CTS ラインの状態を論理値で返します。

解説
CTS ラインの状態を取得します。一般的にはラインステータスが変わった時点で OnCommNotify イベントが発生しますから、このイベント中でどのラインのステータスがどのように変化したのかを取得するためにこのメソッドを使います。

Visual Basic サンプル

```
Sub Comm1_OnCommNotify()
Dim status as integer

statuc = Comm1.GetCTS()
If status = True Then
   Debug.Print “CTS ラインはオンになりました”
Else
   Debug.Print “CTS ラインはオフになりました”
End If
End Sub
```

GetDSR

書式
GetDSR() As Integer

戻り値
DSR ラインの状態を論理値で返します。

解説
DSR ラインの状態を取得します。一般的にはラインステータスが変わった時点で OnCommNotify イベントが発生しますから、このイベント中でどのラインのステータスがどのように変化したのかを取得するためにこのメソッドを使います。

Visual Basic サンプル

```
Sub Comm1_OnCommNotify()
Dim status as integer

statuc = Comm1.GetDSR()
If status = True Then
   Debug.Print “DSR ラインはオンになりました”
Else
   Debug.Print “DSR ラインはオフになりました”
End If
End Sub
```

GetRLSD

書式
GetRLSD() As Integer

戻り値
RLSD[5]の状態を論理値で返します。

解説
RLSD 信号の状態を取得します。

Visual Basic サンプル

```
Sub Comm1_OnCommNotify()
Dim status as integer

statuc = Comm1.GetRLSD();
If status = True Then
   Debug.Print “RLSD はオンになりました”
Else
   Debug.Print “RLSD オフになりました”
End If
End Sub
```

GetRing

[5] receive-line-single-detect

書式
GetRing() As Integer

戻り値
Ring ラインの状態を論理値で返します。

解説
Ring ラインの状態を取得します。一般的にはラインステータスが変わった時点で OnCommNotify イベントが発生しますから、このイベント中でどのラインのステータスがどのように変化したのかを取得するためにこのメソッドを使います。

Visual Basic サンプル

```
Sub Comm1_OnCommNotify()
Dim status as integer

statuc = Comm1.GetRing();
If status = True Then
   Debug.Print "Ringing!"
End If
End Sub
```

OpenComm

書式
OpenComm() As Integer

戻り値
メジャー通信エラー･コード。エラー･コード一覧を参照してください。

解説
VcAutoOpen プロパティが False の場合に、コードで明示的に通信ポートをオープンする場合に使います。以下はサンプル･コードです。戻り値はエラー･イベントに渡される最初のパラメータ(MajorErrorCode)と同じ値です。

Visual Basic サンプル

```
Sub Form1_Load()
Dim rc As Integer
rc = VBManComm1.OpenComm
If rc <> 0 Then
   MsgBox "ポートのオープンに失敗 " & CStr(rc)
End If
End Sub
```

RecvBinaryBytes

書式

RecvBinaryBytes(*RecvData* As Variant, _
Length As Variant) As Integer

パラメータ

RecvData
データを受け取るエリアを Byte 型配列で指定します。

Length
受信する文字列の長さ(バイト)。0を指定した場合は、メソッドを呼び出した時点で受信バッファにあるデータすべてを返します。このパラメータは省略可能で省略した場合は0を指定した場合と同じ動作をします。受信するデータ長を指定した場合は、指定されたバイト数だけ受信するまで、メソッド内部でブロッキングされます。パラメータを省略した場合、または0を指定した場合はメソッド内部ではブロッキングされません。受信データが無い場合は戻り値にERR_NO_DATA ステータスが設定され制御が呼び出し側に戻されます。

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

Visual Basic を使った場合には Byte 型の配列に直接データを受信します。Visual Basic 以外のコントロール・コンテナでは VARIANT 型の1バイトデータ配列としてデータの受信が可能になります。VARIANT 型の扱いについては各コントロール・コンテナの仕様または Wind32API の VARIANT 型の説明や SafeArray 関連 API の説明をご覧ください。

第 2 パラメータは省略することができます。第 2 パラメータを省略した場合または0を指定した場合は、メソッドを呼び出した時点で受信バッファにあるデータより配列サイズが小さい場合は、配列サイズまでデータを読み込み、それを超えるデータについては通信バッファに残されます。

Visual Basic サンプル

```
Dim b(0 To 10) As Byte
Dim rc As Integer, Dim I As Integer

rc = Comm1.RecvBinaryBytes(b)
If rc <> 0 Then
   MsgBox "エラー " & CStr(rc)
   Stop
End If

For I = 0 To 9
   Debug.Print Chr(b(I))
Next I
```

RecvFile

書式

RecvFile(*FileName* As String) As Integer

パラメータ
受信するファイル名。フルパス指定しない場合はカレント･ディレクトリにあるファイルに受信します。

戻り値
メジャー通信エラー･コード。エラー･コード一覧を参照してください。

解説
マルチスレッドによるファイル受信を開始します。パラメータは受信するファイル名です。このメソッドを使う前に VcProtocol プロパティにファイル転送プロトコルを設定します。このメソッドはファイル転送のスレッドを起動してすぐに呼び出し側のユーザー･プログラムに制御を戻します。スレッドの起動に失敗した場合はこのメソッドの戻り値を設定してエラーを返します。ファイル転送中の進捗状況は VcStatus プロパティにあります。ファイル転送が終了すると、OnCommTransferEnd イベントが発生します。

Visual Basic サンプル

```
Sub StartRecvFile()
Dim rc As Integer, fn$
fn$ = InputBox("受信ファイル名を入力してください")
rc = Comm1.ReceiveFile(fn$)
If rc <> 0 Then
   MSgBox "ファイル受信を開始できません。"
End If
End Sub
‘ ファイル転送が終了するとこのイベントが発生する。
Sub Comm1_OnCommTransferEnd( sStatus As Integer )
If sStatus = 0 Then
   MsgBox "ファイル転送が正常に終了しました"
Else
   MsgBox "ファイル転送が異常終了しました" & CStr(sStatus)
End If
End Sub
```

RecvString

書式
RecvString(*SizeToRecv* As Integer) As String

パラメータ
受信する文字列の長さ(バイト)。0を指定した場合は、受信バッファにあるデータすべてを返します。

戻り値
受信した文字列。

解説
文字列を受信します。パラメータは受信する文字数(バイト)です。パラメータに0を指定した場合は受信バッファにデータが無い場合はヌルがかえされます。受信する文字数を指定した場

合、指定した文字数が得られるまで、VBMan コントロール内部でループします。ループしている間は Windows メッセージ処理ができなくなります。無限ループを避けるためには、パラメータ0を指定し、VBA レベルで DoEvents を使ってメッセージを処理しながら、文字列を受信した方が安全です。

Visual Basic サンプル

```
Function RecvNChar( NumOfCharsToRecv As Integer ) As String
Dim r$, Result$
Static SaveResult$                ‘ 前回の呼び出しで通信バッファの残りを保存

Result$ = SaveResult$
SaveResult$ = “”

While Len(Result$) < NumOfCharsToRecv
   r$ = VBManComm1.RecvString(0)
   If r$ <> “” Then
      Result$ = Result$ & r$
   End If
   DoEvents
Wend

‘ 結果を処理
If Result$ > NumOfCharsToRecv Then
  ‘ 通信バッファには要求されたバイト数より多くのデータが存在した。
   RecvNChar = Left$( Result$, NumOfCharsToRecv )
   SaveResult$ = Right$( Result$, Len(Result$) - NumOfCharsToRecv)
Else
   RecvNChar = Result$
End If
End Function
```

RecvUnicodeString

書式

RecvUnicodeString (*Length* As Integer) As String

パラメータ

受信するデータバイト長です。Unicode の場合2バイトで1文字になります。0を指定した場合はこのメソッドが呼び出された時点での受信バッファのデータ全てから Unicode 文字列を生成します。

戻り値

受信した Unicode 文字列。

解説

ライン上に Unicode 形式でデータが送られてきている場合はこのメソッドにて受信することが簡便です。7BIT データについても 16BIT データ(2 バイト)でデータが送信されていることが必要になります。また、奇数バイト SHIFT-JIS データがライン上に流れている場合は従来の

RecvString メソッド等をご利用ください。

SendBinaryBytes

書式
SendBinaryBytes(*lpData* As Variant) As Integer

パラメータ
lpData　　送信するデータ

戻り値
メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説
Visual Basic をお使いの場合にはパラメータで指定された Byte 型のデータを送信します。それ以外のコントロール・コンテナでは VARIANT 型の1バイト配列データの扱いになりますので各コントロール・コンテナのマニュアル等をご参照ください。Win32 API では VARIANT 型と SafeArray 系の API の説明をご参照ください。 送信するデータのサイズは宣言した Byte 型の配列のサイズになります。

Visual Basic サンプル

```
Dim b(0 To 5) As Byte
Dim rc As Integer

b(0) = 0            ‘ Packet size first byte
b(1) = 5            ‘ Packet size second byte
b(2) = 2            ‘ STX
b(3) = AscB(“T”)    ‘ Text body
b(4) = 3            ‘ ETX

rc = VBMComm1.SendBinaryBytes(b)
If rc <> 0 Then
   MsgBox “SendBinaryBytes エラー “ & CStr(rc)
   Stop
End If
```

SendBreak

書式
SendBreak() As Integer

戻り値
メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説
通信ポートをブレーク状態にします。以下はサンプルです。ブレーク状態の解除には ClearBreak メソッドを使用します。戻り値はエラー・イベントに渡される最初のパラメータ

(MajorErrorCode)と同じ値です。

Visual Basic サンプル

```
Dim rc As Integer
rc = VBManComm1.SendBreak
```

SendByte

書式

SendByte(*NumOfChars* As Integer) As Integer

パラメータ

送信データのサイズ。送信バッファ長より大きい値は指定できません。

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説

このメソッドがよびだされた時の送信バッファの内容を指定されたバイト数だけ送信します。Visual Basic 4.0J 32Bit 版以降は String 型が Unicode から Shift JIS に変換される為、0x81 からのシフト JIS コートの先頭バイト・コードが SendBinary メソッドで送信できないことに対応してバージョン 2.00a から追加されたメソッドです。送信するバイナリ・データを送信バッファに設定するには、SendBuffer プロパティを使ってください。

注意

このメソッドは旧バージョンとのコンパチビリティのために残されています。新規にコードを開発する場合は SendBinaryBytes メソッドをお使いください。

Visual Basic サンプル

```
Dim I as integer, rc as integer

‘ SendBuffer はOオフセットでデータ型は Integer です。
For I = 0 To 255
   VBManComm1.SendBuffer(I) = I
Next I
‘ パラメータは送信するバイト数です。
VBManComm1.SendByte(256)
If rc <> 0 Then
   MsbBox “send byte error “ & CStr(rc)
End If
```

SendChar

書式

SendChar(*aChar* As String) As Integer

パラメータ

送信する1文字。

戻り値
メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説
1文字を送信します。指定された文字は、送信バッファの先頭に置かれます。当メソッドの送信結果はトレースツールには表示されません。

Visual Basic サンプル
```
Dim rc As Integer
rc = VBManComm1.SendChar(chr$(13))          ‘ 改行コードを送る
```

SendFile

書式
SendFile(*FileName* As String) As Integer

パラメータ
送信するファイル名。フルパス指定しない場合はカレント・ディレクトリのファイルを指定したものとみなします。

戻り値
メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説
マルチスレッドによるファイル送信を開始します。パラメータは送信するファイル名です。このメソッドを使う前に VcProtocol プロパティにファイル転送プロトコルを設定します。このメソッドはファイル転送のスレッドを起動してすぐに呼び出し側のユーザー・プログラムに制御を戻します。スレッドの起動に失敗した場合はこのメソッドの戻り値を設定してエラーを返します。ファイル転送中の進捗状況は VcStatus プロパティにあります。ファイル転送が終了すると、OnCommTransferEnd イベントが発生します。

参照
ReceiveFile メソッド、OnCommTransferEnd イベント

SendString

書式
SendString(*aStr* As String) As Integer

パラメータ
送信する文字列。

戻り値
メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説
文字列を送信します。ライン上には 7bit キャラクターはそのまま ASCII コードで送信されます。漢字が含まれる場合は Shift-JIS コードにてラインに送信されます。Unicode 文字列をそのままラインに送信したい場合は SendUnicodeString メソッドをご利用ください。バイナリ・データにつきましては SendBinaryByetes メソッドをご利用ください。

以下は、モデムを初期化する文字列を送信する例です。

Visual Basic サンプル
```
Dim s$, rc As Integer

s$ = "ATZ" & Chr$(13) & Chr$(10)
rc = VBManComm1.SendString(s$)
```

SendUnicodeString

書式
SendUnicodeString(*aStr* As String) As Integer

パラメータ
送信する文字列。

戻り値
メジャー通信エラー・コード。
エラー・コード一覧を参照してください。

解説
文字列を送信します。ライン上にも Unicode 形式のままデータを送信します。このメソッドで送ったデータは RecvUnicodeString メソッドで受信することが可能です。7bit 文字でもライン上には2バイト出力されることにご注意ください。Visual Basic 等 32bit 言語では文字列内部コードは Unicode を採用していますが、一般的に周辺機器に漢字データを送る場合は SHIFT-JIS コードになりますので SendString 等のメソッドをお使いください。

以下は文字列を送信する例です。

Visual Basic サンプル
```
Dim s$, rc As Integer

s$ = "ユニコード"
rc = VBManComm1.SendString(s$)
```

SetDTR

書式
SetDTR() As Integer

戻り値

メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説
DTR ラインをオンにします。

SetRTS

書式
SetRTS() As Integer

戻り値
メジャー通信エラー・コード。エラー・コード一覧を参照してください。

解説
RTS ラインをオンにします。

カスタム・イベント

この章では VBMan Control for RS-232C のカスタム・イベントについて解説します。当通信コントロールでは以下のカスタム・イベントがサポートされます。

イベント名	概要
CommError	通信エラーイベント
CommNotify	ライン状況イベント
CommRecv	データ受信イベント(パラメータは文字列)
CommRecvVar	データ受信イベント(パラメータは Variant)
CommSend	データ送信イベント
CommTransferEnd	ファイル転送終了イベント

CommError

VBMan Control for RS-232C 通信エラーの通知にイベントを発生させます。エラー・イベントは以下の形式です。

CommError(*MajorErrorCode* As Integer, *MinorErrorCode* As Long)

MajorErrorCode

主エラーコード。エラーの主たる原因を示します。

MinorErrorCode

詳細情報エラー・コード。送受信時のエラーについて詳細情報が通知されます。

CommNotify

VBMan Control for RS-232C はライン状況が変化した場合にイベントでユーザー・プログラムに通知します。ライン状況イベントに渡されるパラメータはイベントマスク値です。通知されるライン状況は DTR,CTS,RLSD,Ring です。

イベント・マスク値は以下の値です。(16 進表示)

CTS	0x0008
DSR	0x0010
RLSD	0x0020
RING	0x0100

以下はコード・サンプルです。

```
Sub Comm1_CommNotify( dwEventMask As Long )
Dim Stat As Integer
Const EV_CTS = &H8
Const EV_DSR = &H10
Const EV_RLSD = &H20
Const EV_RING = &H100

Select Case dwEventMask
   Case dwEventMask And EV_CTS
      Debug.Print “CTS is On”
   Case dwEventMask And EV_DSR
      Debug.Print “DSR in on”
   Case dwEventMask And EV_RLSD
      Debug.Print “RLSD is On”
   Case dwEventMask And EV_RING
      Debug.Print “Ringing”
End Select

End Sub
```

CommRecv

プロパティ VcNotifyRecvChar に 0 以外の値を設定されており、通信中にこのプロパティの設定値以上に受信バッファにデータが存在する時 CommRecv イベントが発生します。このイベントにはその時点で通信バッファにあるデータがすべて渡されます。以下はサンプル･コードです。Visual C++の場合は通常の ANSI 文字列ではなく BSTR 型の文字列でデータがイベントに渡されることに注意してください。

Visual Basic サンプル

```
Sub Comm1_CommRecv( RecvStr As String )
 Text1.Text = Text1.Text & RecvStr
End Sub
```

Visual C++サンプル

```
void CTestDlg::OnCommRecv(BSTR FAR* strRecv)
{
   CString cstr(*strRecv);
   TRACE(“受信文字列  = %s¥n”,cstr.GetBuffer(0));
}
```

CommRecvVar

プロパティ VcNotifyRecvChar に 0 以上の値を設定されており、プロパティ VcRecvEventType に AsString(=値 0)が設定されている場合に発生するイベントです。通信中に VcNotifyRecvChars プロパティの設定値以上に受信バッファにデータが存在する時

CommRecvVar イベントが発生します。このイベントにはその時点で通信バッファにあるデータがすべて渡されます。イベントのパラメータは Variant 型になっています。この形式のイベントは Visual Basic.NET/Visusl C#等ではサポートされますが、Visual Basic 6.0 等サポートされない言語もありますのでご注意ください。

Visual Basic.NET サンプル

```
Private Sub AxVBMCom2_CommRecvVar(ByVal sender As Object,
   ByVal e As AxVBMCOM32Lib._VBMEvents_CommRecvVarEvent) Handles
       AxVBMCom2.CommRecvVar
   System.Diagnostics.Debug.WriteLine(e.vData)
 End Sub
```

Visual C# サンプル

```
 private void axVBMCom2_CommRecvVar(object sender,
AxVBMCOM32Lib._VBMEvents_CommRecvVarEvent e)
 {
   if(axVBMCom2.VcRecvEventType ==
   VBMCOM32Lib.enumRecvEventType.AsVariantString)
   {
     System.Diagnostics.Debug.WriteLine("recv var " +
                Convert.ToString(e.vData.ToString()));
   }
   else
  {
    System.Diagnostics.Debug.WriteLine("recv varbinary:" +
       e.vData.GetType().ToString() + " ;" + e.vData.ToString());
  }
}
```

CommSend

プロパティ VcNotifySendComplete に True の値が設定されており、送信キューが空になった時に CommSend イベントが発生します。

CommTransferEnd

ファイル転送メソッド RecvFile,SendFile はスレッドをスタートさせるとすぐにユーザー・プログラムに制御を戻します。ファイル転送が終了すると、このイベントに転送の終了が通知されます。イベントにはファイル転送の完了コードが返されます。詳細はエラー･コード一覧を参照してください。

```
Sub Comm1_CommTransferEnd( sStat As Integer )
If sStat = 0 Then
   MsgBox “ファイル転送が正常に終了しました。”
Else
```

```
    MsgBox “ファイル転送が失敗しました。” & Cstr(sStat)
End If
End Sub
```

通信ユーティリティ

VBMan Control for RS-232C には vbmcu32.ocx という通信サポート用の ActiveX Control が添付されます。こちらのカスタム・コントロールも ATL を使って作成されております。この章ではこのカスタム・コントロールによる CRC 計算方法を説明します。

カスタム・メソッド

Crc16

書式

Crc16(*Buf()* As Byte,
Size As Integer,
Result As Integer) As Integer

パラメーター

Buf	Byte 型の配列で指定する CRC16 を計算する領域
Size	上記配列での有効なサイズ。省略した場合は配列サイズになります。
Result	CRC16 計算結果

戻り値

0	正常終了
-1	パラメータ・エラー。配列のサイズ、ディメンション等の指定間違い

概要

16bit CRC を計算します。

サンプル・コード

```
‘
‘以下のサンプルでは Text1 というテキスト・ボックスから入力され
‘たファイルの crc16 を計算して Result というラベルに表示します。
‘
Private Sub Command1_Click()
On Error GoTo err_handler
Dim buf(0 To 1000) As Byte
Dim ch
Dim idx As Integer, crc16 As Integer, rc As Integer, crc32 As Long
```

```
idx = 0
Open Text1.Text For Input As #1
Do While Not EOF(1)                 ' ファイルの終端までループを繰り返します。
   ch = Input(1, #1)                ' 1 文字のデータを読み込みます。
   buf(idx) = AscB(ch)              '
   idx = idx + 1                    '
Loop
Close #1

Debug.Print "file size = ", idx

If optCrc16.Value = True Then
   rc = util.crc16(buf, idx, crc16)
   Result.Caption = Hex$(crc16)
Else
   rc = util.crc32(buf, idx, crc32)
   Result.Caption = Hex$(crc32)
End If

Exit Sub

'-----------------------------
'
err_handler:
MsgBox "Err = " & CStr(Err)
On Error Resume Next
Close #1

End Sub
```

Crc32

書式

Crc32(*Buf()* As Byte,
Size As Integer,
Result As Long) As Integer

パラメーター

Buf	Byte 型の配列で指定する CRC32 を計算する領域
Size	上記配列での有効なサイズ。省略した場合は配列サイズになります。
Result	CRC32 計算結果

戻り値

0	正常終了
-1	パラメータ・エラー。配列のサイズ、ディメンション等の指定間違い

概要
32bit CRC を計算します。

エラー・コード

メジャー・エラー・コード

以下はエラー・イベント・プロシージャに通知される最初のパラメータ(MajroErrorCode部分）の説明です。

ERR_OPEN	100	通信ポートがオープンできません。
ERR_BUILD_DCB	101	Data Control Blockが作成できませんでした。
ERR_COMM_STATE	102	通信状態エラー。詳細はMinorErrorCodeを調べてください。
ERR_NO_MEM	103	メモリが不足しています。
ERR_BUFFER_SHORT	104	受信するキューのサイズが小さい。 キュー・サイズを大きくしてください。
ERR_READ_COMM	105	通信ポートから読み込めません。 詳細はMinorErrorCodeを調べてください。
ERR_WRITE_COMM	106	通信ポートに出力できません。 詳細はMinorErrorCodeを調べてください。
ERR_CLEAR_BREAK	107	ブレーク状態をクリアできません。
ERR_SET_BREAK	108	ブレーク状態に移行できません。
ERR_TRANSMIT_CHAR	109	SendCharに失敗しました。
ERR_INVALID_SIZE	110	SendCharで文字列が指定されました。
ERR_NOT_OPEN	111	通信ポートがオープンしていません。
ERR_ALREADY_OPEN	112	2度通信ポートのオープンを試みました。
ERR_INVALID_DEVICE_NAME	113	通信ポートの指定が不正です。
ERR_FLUSH_COMM	114	通信キューの廃棄に失敗しました。
ERR_RECV_TIMEOUT	115	受信タイムアウト
ERR_CREATE_EVENT	116	イベントの作成に失敗しました。
ERR_RECV_LENGTH_TOO_LONG	117	RecvStringメソッドへのパラメータが大きすぎます。
ERR_THREAD	118	スレッドの作成に失敗しました。
ERR_CLEAR_DTR	119	DTRのクリアに失敗しました。
ERR_SET_DTR	120	DTRラインをセットできません。
ERR_CLEAR_RTS	121	RTSラインをクリアできません。
ERR_SET_RTS	122	RTSラインをセットできません。

ERR_GET_MODEM_STATUS	123	モデムの状態を取得することに失敗しました。
ERR_COMM_LINE	124	ライン状態を取得することに失敗しました。
ERR_IN_TRANSFER	125	ファイル転送中です。
ERR_NOT_IN_TRANSFER	126	ファイル転送中ではありません。
ERR_SEND_TIMEOUT	127	送信タイムアウトです。
ERR_TYPE_INVALID	128	Byte型の配列を指定してください。また指定するByte型の配列は1次元のみ指定可能です。
ERR_NO_DATA	129	RecvBinaryBytesメソッド等の呼び出し時点で通信バッファにはデータが存在しませんでした。

マイナー・エラー・コード

エラー・イベント・プロシージャに通知される2番目のパラメータはGetLastError APIからの戻り値です。エラーの詳細はマイクロソフトWin32SDKのドキュメント等を参照してください。エラーの定義はWin32 SDKにあるヘッダー・ファイルwinerror.hにあります。

ファイル転送ステータス

以下はファイル転送のステータスです。CommTransferEnd カスタム・イベントのパラメータに設定される値です。

シンボル	値	意味
ERR_XMODEM_FILE_EXSIST	2101	バイナリ・ファイル受信で指定したファイルがすでに存在します。
ERR_XMODEM_FILE_OPEN	2102	ファイルをオープンできません。
ERR_XMODEM_FILE_READ	2103	ファイルの読み込みエラー。ディスクまたはファイルシステムが破損しています。chkdsk,scandisk でディスクを検査してください。
ERR_XMODEM_FILE_WRITE	2104	ファイルの書き込みエラー。ディスクまはたファイルシステムが破損しているか、ファイルシステムに空領域が無いと思われます。chkdsk,scandisk でディスクを検査してください。

ERR_XMODEM_SEND_CHAR	2105	文字を送信できません。
ERR_XMODEM_SEND_SHORT	2106	CRC またはシーケンス送信エラー
ERR_XMODEM_SEND_PACKET	2107	パケットを送信できません。
ERR_XMODEM_RECV_CHAR	2108	文字受信エラー
ERR_XMODEM_RECV_SHORT	2109	CRC またはシーケンス受信エラー
ERR_XMODEM_RECV_PACKET	2110	パケット受信エラー
ERR_XMODEM_SEQ	2111	シーケンス番号に誤り
ERR_XMODEM_PROTOCOL	2112	プロトコル指定が誤り
ERR_XMODEM_RETRY_OUT	2113	リトライ・エラー
ERR_XMODEM_ABORT_TRANSFER	2114	ファイル転送の中断

トラブルシューティング

ここでは VBMan Control for RS-232C を使ってアプリケーションを開発する場合に多く発生するトラブルについての解決方法を記述します。最新の情報、追加情報につきましてはテクナレッジのサポート web をご参照ください。URL は以下になります。

http://www.techknowledge.co.jp/techinfo.html

USB シリアルアダプタでの利用

弊社で RS-232c 機器を USB ポートから接続するための変換アダプタをいくつかテストしました。問題なくご利用いただけることを確認しております。最近の PC では 1 ポートしかサポートしない機種も多いので、複数ポートをご利用になる場合に便利だと思います。弊社で動作確認したのは PCI コミュニケーション社の URS-03 とトライコーポレーション JUSTY UCR-01 です。

Xmodem で転送中に進捗状況を表示したい

Xmodem プロトコルでは転送するデータ・サイズをあらかじめ交換しないため、何パーセント転送済みなのかを知ることはできません。アプリケーションでパーセント表示が必要な場合はあらかじめ転送するファイルサイズを交換するプログラムを作成して、このプロパティからパケット数を得て計算してください。ファイル転送されたパケット数は VcProgress プロパティに保持されていますので、X-Modem プロトコルの場合はこれに 128 バイトのパケットサイズを乗算するとファイル転送したサイズを求めることができます。また、X-Modem 1K の場合は 1024 を乗算します。

VC++ 6.0 DeveloperStudio の全プロパティ表示

上記ダイアログでは日本語版にもでも英数フォントを使っているため、プロパティの選択枝に漢字が含まれるものが化けて表示されます。 Developer Studio の日本語対応が不十分なことが原因と思われます。表示化けするプロパティは VBMan のプロパティ・ダイアログで設定してください。

Access97 でプロパティが保存されない

Access97 のフォームで当コントロールをご利用になる場合はプロパティ値としてデフォルト値以外を設定する場合は、Form_Load でコードで各プロパティを設定してから OpenComm メソッドを呼び出してください。Samples¥Access フォルダーに Bcode.mdb にサンプルがありますのでご参照ください。

VB5 でプロパティ・ダイアログの「適用」ボタンがイネーブルされない。

VB4 等他のコントロール・コンテナでは問題無いので VB5 で ATL コントロールを使った場合

の不具合と判断しています。OK ボタンにてプロパティ値を設定してください。

Delphi/C++Builder でプロパティ・ダイアログに選択枝がドロップ・ダウンしない。

ATL コントロールを使った場合の Delphi/C++Builder の制約と思われます。選択枝が表示されないプロパティはプロパティ・ダイアログで設定するか、実行時コードでプロパティを設定してください。

インストールでエラーが発生する。または、regsvr32.exe にて ActiveX コントロールを登録できない。

ATL(ActiveX Control Library)を使って開発された ActiveX Control は OLEAUT32.DLL のバージョン 2.20.4049 以降が必要になります。Visual Basic, Microsoft Internet Explorer 等をインストールすると OLEAUT32.DLL がアップデートされます。OLEAUT32.DLL のバージョンはマイクロソフト・システム情報(msinfo32.exe)等で確認することができます。

拡張ボードでの動作

VBMan Control for RS-232C は各種拡張ボードでの動作確認は基本的にしておりません。VBMan のソフトウェア構造としては Win32 API を呼び出して通信ドライバーとデータを交換します。ハードウェアを直接制御するようなことはありません。通信ポートとして COM1 から COM10 まで指定できるようになっていますが、これらの指定は Windows/Win32 API へポートのオープン時に一度指定するだけです。従って、各種拡張ボードでの動作はご利用になられる OS でメーカー様から提供されるデバイス・ドライバーが Win32 API を経由して正常に動作するものでしたら VBMan Control for RS-232C でも問題なくご利用いただけます。

Visual Basic セットアップ・ウィザードでのモジュール配布

VBMan Control for RS-232C を使ったアプリケーションを Visual Basic のセットアップ・ウィザードを使って、 インストール・プログラムを作成する場合、以下の行を Windows のディレクトリにある swdepend.ini ファイルに追加することで、 配布プログラムに OCX を自動的に追加することができます。 OCX のファイル名を[]の中に記述して以下の5行を追加することになります。セットアップ・ウィザード(7/7)で「モジュールの追加」で OCX を指定すれば、以下の設定は不要です。 実行時のモジュールとして vbmshm32.dll を同ウィザードにて指定することも必須です。

```
[VBMCOM32.OCX]
Register=$(DLLSelfRegister)
Dest=$(WinSysPath)
Uses1=OCX Runtime Support
Uses2=
```

NEC 98 シリーズをお使いの場合

NEC98 シリーズでは、ハードウェアの違いから、シリアル関連のドライバーが IBM 互換機とは別のものが用意されてる機種があります。通信ドライバーのバージョンによっては、正常に通信できないものが確認されています。NEC のイフォーメーション、Nifty Serve の FNECINFO, 同社の WEB サイトなどで修正モジュールが配布されていますので、最新版のドライバーを導入して動作をご確認ください。

サポート WEB サイトから新バージョンをダウンロードしてインストールしたら動作が変わった

サポート WEB サイト(www.techknowledge.co.jp)では、VBMan Control for RS-232C の最新バージョンがダウンロード可能になっています。最新版をダウンロードして利用する場合には、¥Windows¥system 等にある vbmcom32.oca ファイルを削除して、regsvr32.exe で vbmcom32.ocx を再度登録してください。拡張子.oca のファイルは OCX のプロパティ情報などをキャッシュしているファイルです。プロパティを追加した新バージョンをインストールするとタイプ情報の不一致から動作が不安定になることがあります。また、古いバージョンの OCX で作成された EXE ファイルは再度コンパイルが必要になる場合もあります。再度コンパイルするために開発モードで実行した Visual Basic ではフォームを一旦ひらいて vbmcom32.ocx のプロパティをご確認ください。正常な値が設定されていない場合はコントロールをはり直して、正確なプロパティ値を設定してください。

Access97 でのバイナリ・データ通信

Access97 では Byte 型の配列操作に不具合がありバイナリデータを SendBinaryBytes, RecvBinaryBytes メソッドでは送受信することができません。回避するための方法として RecvBuffer/SendBuffer プロパティ、SendByte メソッドをお使いください。

以下はコードサンプルです。

1.RecvBuffer プロパティについて
RecvBuffer を参照することによって受信データのバイナリ値を
Integer 型(2 バイト整数)の変数に取得することが可能です。
プロパティは一次元配列でインデックスは 0 からとなります。

```
Dim tmp$, v as Integer, i As Integer

tmp$ = Com.RecvString(0)
If tmp$ <> "" Then                    ' 受信データ有？
   For i = 0 To Len(tmp$) - 1
      v = Com.RecvBuffer(i)
      Debug.Print Cstr(v)
   Next i
End If
```

2.SendBuffer プロパティと SendByte メソッド
SendBuffer プロパティにセットした送信データは SendByte メソッド
で送信することができます。

```
Dim rc As Integer, i As Integer
For i = 0 to 255
   Com.SendBuffer(i) = i
```

```
Next i
rc = Com.SendByte(256)
If rc <> 0 Then
   MsgBox "SendByte エラー " & CStr(rc)
End If
```

Access97 等の VBA では&H20 以下のバイナリ・データは SendString/RecvString メソッドで送受信可能です。&H80 以上のデータ通信が無い場合は SendString/ RecvString メソッドを使ったほうが、コードが簡略化される場合があります。尚、RecvString メソッドで受け取った文字列のバイナリ値を知る場合には AscB 関数を使ってください。

付録

シリアル通信

パーソナルコンピュータは、外部と通信するために、通常2種類の I/O ポートを備えています。一つは、モデムを使った通信に利用するシリアルポートで、もう一つは、プリンタとの接続に使うパラレルレポートです。

シリアルポートは、１本の線を使って1ビットずつ送受信するので、ビットデバイスと呼ばれます。ビットデバイスは、同じ情報を送るのにバイトデバイスの8倍の時間が必要ですが、２～３ほんの先からなる安価なケーブルを使える利点があります。実際、双方向通信に必要なのは、送信用、受信用、接地用の3本だけです。

双方向通信

双方向通信には、半二重方式と全二重方式があります。半二重方式は、データを双方向に送りますが、送信中には受信が、また受信中には送信ができません。半二重方式は、モデム間の通信方式としてよく使われます。全二重方式は、送信しながら同時に受信もできる方式です。コンピュータのシリアルポートは全二重方式を採用しており、送信と受信には別の線を使います。1つの回線で全二重通信をサポートしているモデムもあります。

双方向通信のほかに、データを一方方向にしか送信できない単方向通信があります。これは最も単純な通信方式で、端末は受信専用、ホストは送信専用として働きます。パラレルプリンタポートでは、コンピュータからプリンタに一方的にデータを送るだけなので、この方式を採用しています。

スタートビットとストップビット

非同期通信でデータを送る時は、データビットの前後にスタートビットとストップビットを送信します。データビット長は5、６、7または8ビットに設定します。送信側と受信側は、スタートビットとストップビットのタイミングと同様に、このデータビット長も合わせる必要があります。

データビット長を7ビットにすると、127 以上の ASCⅡコードは送ることができません。5ビットでは、最高でも３１までの ASCⅡコードに制限されます。データビットに続いて送信するストップビットの値は1（マーク状態）で、直前のビットの値が1でも、この値は正しく検出されます。なお、ストップビット長は1、1.5、2ビットのいずれかに設定します。

パリティビット

パリティビットは、転送中に生じた誤りを検出するためのもので、データビットとストップビットの間に挿入します。

このパリティビットは、データビット中のマーク状態（値が1）の数が偶数か奇数かを1ビットで表します。パリティには、マーク状態が偶数個の時にパリティビットの値を0にする偶数パリティと、奇数個のときに値を0にする奇数パリティがあります。例えば、偶数パリティを選択する

と、データ 0110011 のパリティビットはOになり、データ 11010110 のパリティビットは1になります。

パリティビットを使った誤り検出は、完全なものではありません。1ビットの誤りは検出できますが、ビット誤りが遇数個（例えば、値1のビット2個を値0として誤って受信した時）あれば、検出できません。また、パリティビットは、誤りを検出するだけで訂正することはできません。

フロー制御

シリアルデータの場合、データは連続して送信側から受信側へ送られます。受信したデータは、直ちに読み取らなければなりません。読み取る前に次のデータが到着すると、直前のデータが失われてしまうからです。そこで、受信したデータを読み取るまでの間、受信バッファにデータを保存します。これにより、データを受信してから読み取るまで時間に余裕ができるので、データ受信中に別の処理を実行できるようになります。

受信バッファをメモリに割り当てたり、受信バッファにデータを読み込んだりするのは、通信ソフトです。バッファにデータを書き込む速さよりも通信ソフトがデータを読む速さの方が遅いと、バッファはすぐに一杯になり、その後に受信するデータはすべて失われます。そこで、受信バッファが一杯になった時は、シグナルを送って送信を停止し、受信バッファが空いてから再びシグナルを送って送信を再開します。このシグナルのやり取りをハンドシェークと呼び、ハンドシェークを使ってデータの流れを調整することをフロー制御といいます。

RS－232Cインタフェース

RS－232C の”RS”は標準仕様(Recommend Standard)を意味します。また、”232”は標準仕様の認識番号で、”C”はその標準仕様の最新版であることを表しています。大部分のコンピュータのシリアルポートは RS-232C に準拠しています。RS-232C は 25 ピンの”D”コネクタ（そのうち22ピンを使用）を使うことになっています。しかし、ほとんどのピンはパーソナルコンピュータ間の通信に必要ないので、最近では 9 ピンのコネクタがよく使われます。

VBMan Control for RS-232C 調査依頼

以下の調査依頼フォームをコピーし、必要事項を記入してユーザー・サポートまでファックスまたは、同様の項目を記入してインターネットでsupport@techknowledge.co.jp宛メールしてください。折り返し担当者が技術サポートの連絡をさしあげます。もうしわけありませんが**電話によるサポートは受け付けておりません。ご了承ください。**

日付	
会社名	
登録ユーザー名	
製品シリアル番号	
電話番号	
ファックス番号	
メール・アドレス	
使用パソコン機種	
接続デバイス	
OSとバージョン	
言語とバージョン	
お問合わせ内容、問題記述など、具体的に再現可能なようにご記入ください。	
添付資料	

VBMan Control for RS-232C version 4.00

マニュアル第5版
2018年11月29日　第3刷発行

Printed In Japan