

UDPOCX

WILL

株式会社ウィル

白紙ページ

白紙ページ

- Microsoft、Windows、Windows NT、Visual Basic、ActiveX、Office、Access、Excel は、米国 Microsoft Corporation の米国ならびに各国における登録商標です。
- その他本書に掲載されている会社名、製品名はそれぞれ各社の商標又は登録商標です。

目次

はじめに.....	3
商品に含まれるもの.....	5
動作環境について.....	5
インストール.....	6
ライセンスの登録.....	8
サンプルを見る.....	11
サポートについて(無償).....	12
バージョンアップについて(無償).....	13
再配布について.....	15
プログラミング概要.....	17
通信を開始する.....	19
データの送信.....	19
データの受信.....	19
通信の終了.....	19
ブロードキャスト.....	20
その他.....	20
状態遷移図.....	21
プロパティ.....	25
AnyData プロパティ.....	27
Broadcast プロパティ.....	28
Copyright プロパティ.....	29
LastError プロパティ.....	30
LocalHost プロパティ.....	31
LocalAddress プロパティ.....	32
LocalIp プロパティ.....	33
LocalPort プロパティ.....	34
Sendable プロパティ.....	35
Socket プロパティ.....	36
State プロパティ.....	37
UserData1~5 プロパティ.....	38
UserFlag プロパティ.....	39
WinsockVer プロパティ.....	40
WinsockDesc プロパティ.....	41

WinsockSysStat プロパティ.....	42
WinsockMaxSockets プロパティ.....	43
WinsockMaxDatagram プロパティ.....	44
メソッド.....	45
Open メソッド.....	47
SendTo メソッド.....	48
Close メソッド.....	49
htonl メソッド.....	50
htons メソッド.....	51
ntohl メソッド.....	52
ntohs メソッド.....	53
HostIp メソッド.....	54
イベント.....	55
Ready イベント.....	57
Received イベント.....	58
Sent イベント.....	59
WsError イベント.....	60
Winsock エラーコード.....	61
サンプル.....	65
TFTP.....	67
Bootpd.....	68
Syslogd.....	69
timed.....	70
CHAT de UDP.....	71
索引.....	73

はじめに

はじめに

白紙ページ

商品に含まれるもの

1. CD-ROM
 - Willware.exe
 - Cryptdll.exe
(暗号 DLL 専用・実行環境用セットアップキット)
 - readme.txt
2. フロッピーディスク
 - レジストリファイル
 - readme.txt
3. 使用許諾契約書
4. マニュアル

動作環境について

■対応 OS

UDPOCX は、以下に示す OS で動作確認を行っております。

Microsoft Windows 95、Microsoft Windows 98、
Microsoft WindowsNT 4.0、Microsoft Windows 2000、
Microsoft Windows XP、Microsoft Windows 2003

■開発に必要なソフトウェア

UDPOCX をご使用いただくには、以下のいずれかのソフトウェアが必要です。

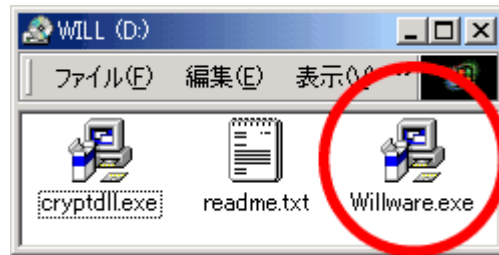
Microsoft Visual Basic Ver 5.0
Microsoft Visual Basic Ver 6.0
Microsoft Office 2000 (Access、Excel)

UDPOCX は、Microsoft Visual C++ Ver 5.0 で作成しています。サンプルは、Microsoft Visual Basic Ver 5.0 で作成しています。

※ 本製品は日本語環境のみの対応となります。

インストール

製品の CD-ROM に含まれているセットアップキット (Willware.exe) をダブルクリックします。

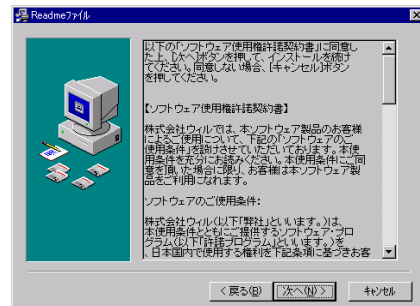


画面にしたがって、インストールを進めて下さい。

1. インストールを始めます。「次へ」をクリックして下さい。



2. 使用許諾契約書です。内容に同意される場合は「次へ」をクリックして下さい。



3. インストール先のフォルダを指定します。初期設定でよろしければ「次へ」をクリックして下さい。別のフォルダを指定したい場合は「参照」をクリックし、フォルダを指定して下さい。



4. インストール中に置換されるファイルのバックアップを作成できます。そのバックアップファイルの保存先フォルダを指定します。初期設定でよろしければ「次へ」をクリックして下さい。



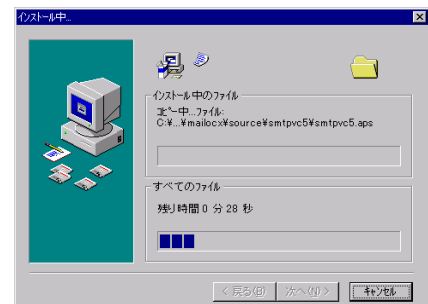
5. WILLWARE Components を登録するスタートメニュー又はプログラムマネージャのグループフォルダを指定します。初期設定では、新規に「WILLWARE Components」の名前でフォルダを作成します。特に指定する必要がなければ、初期設定をお勧めします。



6. プログラムのコピーを開始します。「次へ」をクリックして下さい。



7. プログラムのコピーをしています。中断する場合は、「キャンセル」をクリックして下さい。



8. インストールが完了しました。「完了」をクリックし、インストールを終了して下さい



はじめに

ライセンスの登録

■レジストリファイルから登録する

ライセンスを登録します。製品に含まれているフロッピーディスクのレジストリファイル (EUXXXXXXXXX.reg) をダブルクリックして下さい。(「XXXXXXXXXX」は、任意の数字がファイル名として付けられています。)

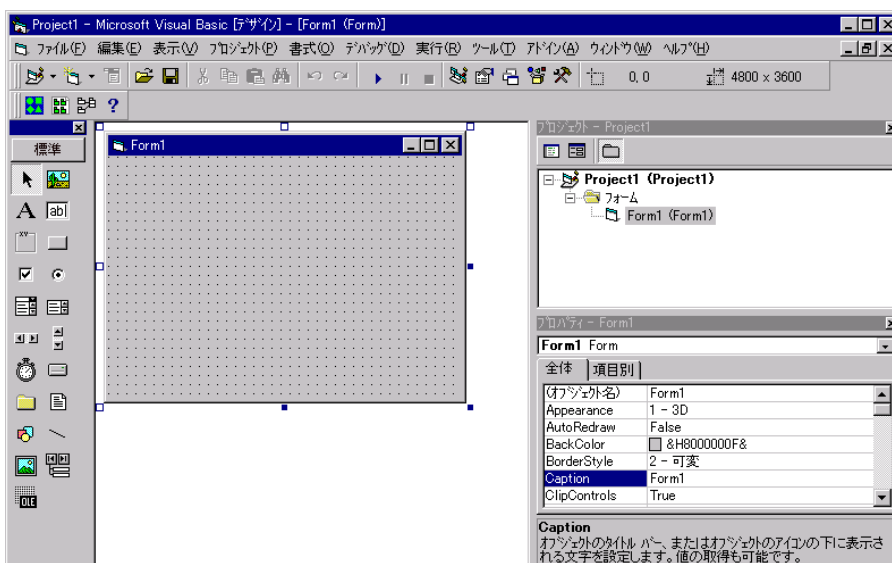


以下のメッセージボックスが表示され、ライセンスがレジストリに登録されます。

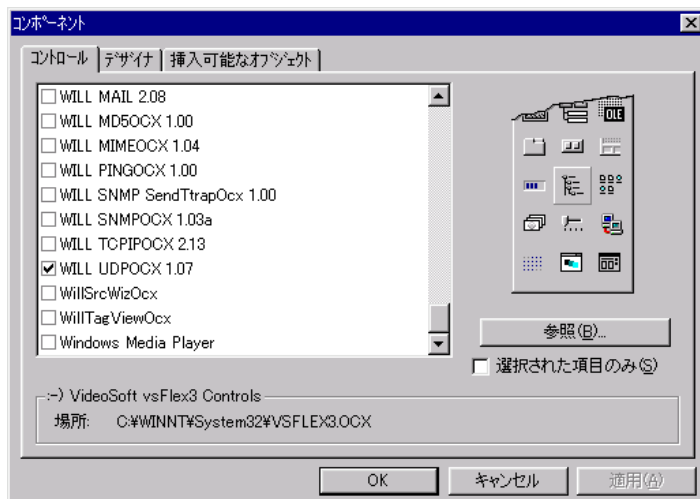


■手動で登録する

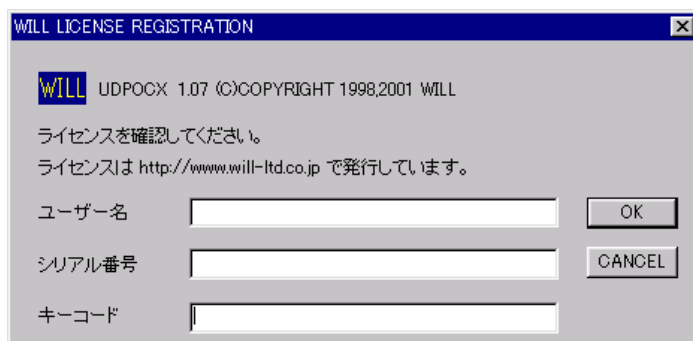
あらかじめ電子メールで通知しているライセンス情報を利用してライセンスを登録する等、レジストリファイルを利用しない場合は、VisualBasic 起動後に新規プロジェクトを選択し以下のデザイン画面を開きます。



ツールバーの「プロジェクト」から、「コンポーネント」を選択し、「コンポーネント」画面を開きます。次にコントロールタブの一覧から UDPOCX を選択して「OK」をクリックすると、UDPOCX がツールボックスに追加され、アイコンが表示されます。



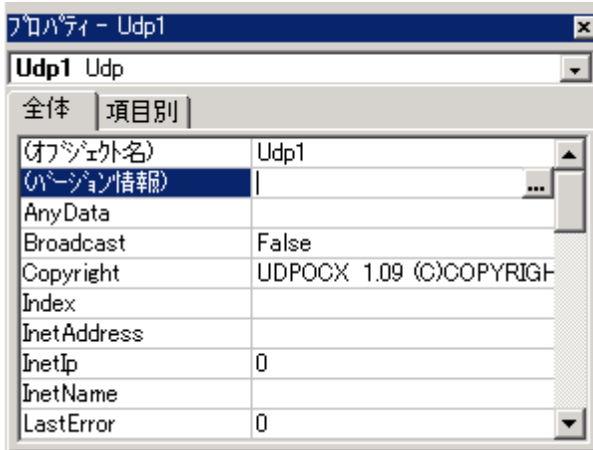
ツールボックスに追加された UDPOCX を選択し、フォームにアイコンを貼り付け、プロパティの「(バージョン情報)」をクリックすると、以下の「WILL LICENSE REGISTRATION」画面が表示されます。ここで、ユーザー名、シリアル番号、キーコードをそれぞれ入力してライセンス登録を行います。



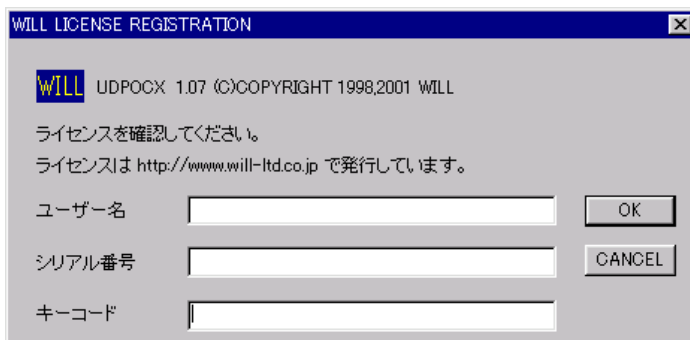
はじめに

■トライアルライセンスから正規ライセンスへの移行

既にトライアルライセンスが登録されている場合には、デザイン画面にある UDPOCX のプロパティで「(バージョン情報)」をクリックして下さい。



「WILL LICENSE REGISTRATION」画面が表示されますので、ここで正規ライセンスを入力して下さい。



■ライセンス入力時のご注意

※ライセンスが入力できない!?

入力したライセンスにスペースが含まれていないか確認して下さい。(ライセンスに、スペースは使用していません。)

※登録したライセンスを認識しない!?

ライセンスを登録しても、オブジェクトが新規ライセンスを認識していない場合は、UDPOCX のアイコンを少し動かして下さい。この作業により、オブジェクトにライセンスが記憶されます。

※トライアルライセンスで作成したアプリケーションはどうする!?

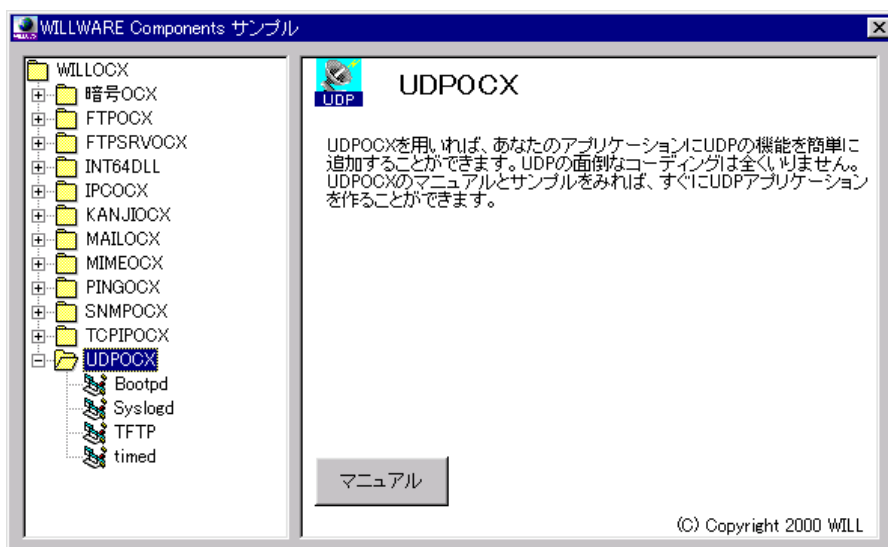
既にトライアルライセンスで作成したアプリケーションは、正規ライセンスを登録した後、再コンパイルする必要があります。

サンプルを見る

インストールが完了すると、スタートメニューに「WILLWARE Components」が追加されます。



「WILLWARE Components」の「サンプル」を起動すると「WILLWARE Components サンプル」画面が表示されます。サンプルの起動、またはそれぞれのソースを開くことができます。但し、ソースを開くにはライセンスが必要です。トライアルライセンス又は、正規ライセンスを登録してご利用下さい。(ライセンスの登録方法は前項の「ライセンスの登録」をご覧下さい。)



はじめに

サポートについて(無償)

サポートは基本的に電子メールで受け付けております。サポートは無償でご利用いただけます。

■お問い合わせの前に

サポート作業を円滑に行うために、お問い合わせの際には以下の情報をご用意下さい。

1. 製品名及びバージョン
2. 開発環境(OSの種類及びバージョン、サービスパッケージの種類)
3. 開発ツール及びバージョン
4. サーバの種類
5. 問題点
 - (1) エラー内容又は、エラー状況のハードコピー。
 - (2) 問題点となる部分のサンプルソースコード。

■FAQ

弊社ホームページの「サポート」のページで、キーワードを入力して FAQ を検索できます。休業日などサポートの対応が遅れる場合もありますので、まずはこちらをご確認下さい。

■お問合せ先

info@will-ltd.co.jp

バージョンアップについて(無償)

製品のバージョンアップは、すべて無償です。

■バージョンアップ情報の入手方法

バージョンアップの情報は、弊社ホームページの新着情報で通知し、各商品のページの更新履歴で更新内容を掲示致します。

■最新バージョンの入手方法

最新バージョンのプログラムは、弊社ホームページ(<http://www.will-ltd.co.jp/>)のダウンロードのページよりダウンロードすることが出来ます。ダウンロードするファイルは、以下のバージョンアップの目的により異なりますのでご注意ください。

- **WILLWARE Components(全製品用)セットアップキットを利用してバージョンアップ**
ファイル名 : 「Willware.exe」

WILLWARE Components(全製品用)セットアップキットは全ての製品をインストールするためのものです。そのため本製品以外の製品及びサンプル、マニュアルも同時にバージョンアップされます。

- **各コンポーネント毎のセットアップキットを利用してバージョンアップ**
ファイル名 : 「○○○ocx.exe」

各コンポーネントのファイル(ocx、dll)及び、依存ファイルのみバージョンアップされません。サンプル及びマニュアルはバージョンアップされませんのでご注意ください。

はじめに

■バージョンアップをする前に

各セットアップキットを利用してバージョンアップをする前に、以下のことにご注意ください。

● WILLWARE Components(全製品用)セットアップキットを利用してバージョンアップする場合は、古いバージョンをアンインストールしてから、最新バージョンをインストールすることをお勧めいたします。

※ アンインストールの方法は、スタートメニューから「設定」→「コントロールパネル」→「アプリケーションの追加と削除」の画面で、「WILLWARE Components」を選択し、画面の指示に従って行って下さい。

● 各コンポーネント毎のセットアップキットを利用してバージョンアップする場合は、最新バージョンをそのままインストールして下さい。古いファイルは上書きされます。

※ 弊社製品を複数ご利用いただいている場合、いずれか1つをバージョンアップしても他の製品に影響はありません。

■バージョンアップの方法

セットアップキットをダブルクリックし、画面の指示に従ってインストールを進めて下さい。

再配布について

■作成したアプリケーションの配布時

UDPOCX を利用して作成したアプリケーションの配布時のランタイムライセンスはフリーです。但し、開発ライセンスの配布はできません。

■再配布時に必要な配布可能ファイル

UDPOCX を利用して作成したアプリケーションを配布する場合には、以下のファイルを添付する必要があります。()内は推奨バージョンです。

- ・ UDPOCX
- ・ MFC42.DLL (Ver 4.21.7022)
- ・ MFC42LOC.DLL (Ver 4.21.7022)
- ・ MSVCRT.DLL (Ver 5.00.7022)
- ・ OLEPRO32.DLL (Ver 5.0.4118)
- ・ OLEAUT32.DLL (Ver 2.20.4118)

※ セットアップウィザードを使用する場合

UDPOCX をインストールすると、自動的に OCX の依存ファイルが以下のディレクトリにインストールされます。

C:\Windows\system (Windows95, Windows98 の場合)

C:\WINNT\system32 (WindowsNT4.0, Windows2000, Windows2003 の場合)

C:\Windows\system32 (WindowsXP の場合)

セットアップウィザードを実行すると自動的にアプリケーション配布時に必要な OCX (内部で利用している OCX)と、DLL ファイルが Setup.lst ファイルに追加されます。

■著作権

- ・ UDPOCX およびこれに付随するマニュアルの著作権は株式会社ウィル(横浜市保土ヶ谷区)にあります。
- ・ 本ソフトウェアおよびマニュアルを運用した結果については、当社は一切責任を負いません。
- ・ 本ソフトウェアの仕様またはマニュアルに記載されている事項は予告無く変更することがあります。
- ・ マニュアルなどに記載されている会社名、製品名は、各社の商標および登録商標です。
- ・ UDPOCXを利用するアプリケーションはUDPOCXの著作権表示を行わなければなりません。Copyright プロパティに UDPOCX の著作権を示す文字列があります。アプリケーションまたはドキュメントのいずれかにこの文字列を表示して、UDPOCX を使用していることを示してください。

はじめに

白紙ページ

プログラミング概要

白紙ページ

通信を開始する

通信を開始するには、Open メソッドを使って送受信を行う準備をします。Open メソッドには、引数が必要です。クライアントとして動作する場合は、引数は“0”を指定します。サーバーとして動作する場合は、ポート番号を示す番号またはサービス名を指定します。たとえば、UNIX の SNMP サーバーに問い合わせを行うには、“0”を指定します(UDPOCX をクライアントとして利用する)。また、SNMP サーバーを UDPOCX を用いて構築するには、“161”を指定します(UDPOCX をサーバーとして利用する)。Open メソッドに成功すると、Ready イベントが発生します。

データの送信

Ready イベントが発生したら、送受信が可能になります。データを送信したい場合は、SendTo メソッドを使用します。SendTo メソッドには送信先の IP アドレス、ポート番号、送信データを渡します。連続してデータを送信することはできません。複数のデータを送信するには、送信後 Sent イベントの発生を待たなくてはなりません。

データの受信

データが到着すると Received イベントが発生します。Received イベントの引数には、相手先の IP アドレス、ポート番号、データがはいっています。データを返信する場合はこの IP アドレスおよびポート番号を使用します。UDPOCX をサーバーとして利用する場合は、ここで新しい UDPOCX オブジェクトを作成し、それ以降の送受信をその新しいオブジェクトに任せます。もちろん、1 つのオブジェクトでサーバーを構成することも可能ですが、通常かなり複雑な処理が必要になります。

通信の終了

通信を終了する場合は、Close メソッドを使用します。Close メソッドを発行すると Closed イベントが発生します。Closed イベントはエラーのときも発生します。

ブロードキャスト

UDP では同一セグメント内でのブロードキャストが可能です。ブロードキャストを行うには、BroadCast プロパティを True にしてください。これを行わないと、ブロードキャストアドレスを指定しても、ブロードキャストとは扱われません。

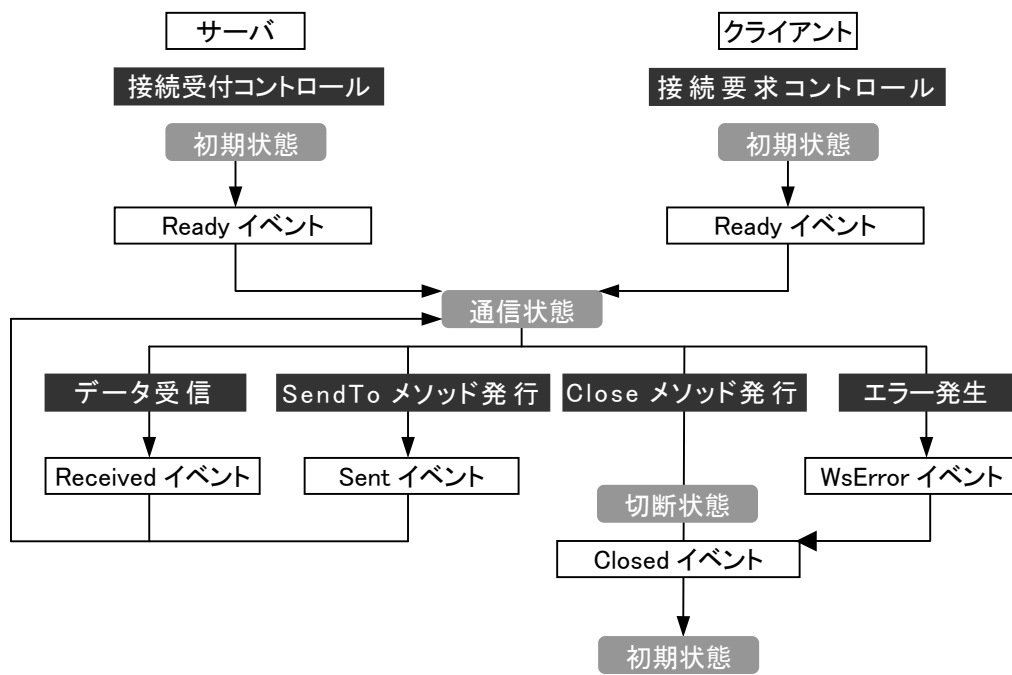
その他

UDP は 512 バイトから 65535 までの大きさのパケットを扱えます。上限はシステムによって異なりますので、汎用的なサーバーを作る場合は、512 バイトのパケットになるように設計してください。

状態遷移図

状態遷移図

白紙ページ



状態遷移図

白紙ページ

プロパティ

プロパティ

白紙ページ

AnyData プロパティ

■機 能

任意のデータを保持します。

■構 文

Object.AnyData[=Value]

AnyData プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOGX オブジェクトです。
Value	任意のデータです。

■データ型

バリエーション型(Variant)

Broadcast プロパティ

■機能

ブロードキャスト送信をするかどうか指定します。True にすると、ブロードキャスト送信を行います。

■構文

Object.Broadcast[=Value]

Broadcast プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	UDPOCX オブジェクトです。
Value	ブロードキャスト送信するかどうかを指定します。次の設定値を参照してください。

■設定値

Value 設定値は次の通りです。

(値)	(説明)
True	ブロードキャスト送信をします。
False	ブロードキャスト送信しません。

■データ型

ブール型(Boolean)

Copyright プロパティ

■機 能

UDPOCX のバージョンと著作権情報です。UDPOCX を用いるアプリケーションは、これを画面又はドキュメントに表示しなければなりません。この値は参照のみ可能です。

■構 文

Object.Copyright

Copyright プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

LastError プロパティ

■機 能

Winsock で何らかのエラーが発生したとき、Winsock の返すエラーコードが格納されます。値の意味は巻末の「Winsock のエラーコード」を参照して下さい。このプロパティはエラーが発生すると上書きされます。

■構 文

Object.LastError

LastError プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

長整数(Long)

LocalHost プロパティ

■機 能

実行中のマシンのホスト名が格納されています。参照のみ可能です。

■構 文

Object.LocalHost

LocalHost プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

LocalAddress プロパティ

■機 能

ローカルの IP アドレスを指定します。通常は指定する必要はありません。ネットワークカードが 2 枚以上刺さっている場合に、特定のカード上のネットワークで通信したい場合には必ず指定しなければなりません。

■構 文

Object.LocalAddress

LocalAddress プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

LocalIp プロパティ

■機 能

接続が確立した時点での、ローカルホストの IP アドレスが格納されます。参照のみ可能です。

■構 文

Object.LocalIp

LocalIp プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

長整数(Long)

LocalPort プロパティ

■機 能

接続が確立した時点での、ローカルホストのポート番号が格納されます。参照のみ可能です。

■構 文

Object.LocalPort

LocalPort プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

整数(Integer)

Sendable プロパティ

■機能

UDPOCX が送信可能状態かどうかを示します。送信可能状態であれば、True が、そうでなければ、False が格納されます。参照のみ可能です。

■構文

Object.Sendable

Sendable プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	UDPOCX オブジェクトです。

■参照値

参照値は次のとおりです。

(値)	(説明)
True	通信可能です。
False	通信不可能です。

■データ型

ブール型(Boolean)

■解説

送信処理が完了するには、時間がかかります。Send メソッドを用いてデータを送信すると、Sendable は False となり、送信が完了して Sent イベントが発生する直前に、Sendable は True になります。

Socket プロパティ

■機能

ソケットハンドルを格納します。-1 のときはソケットが無効状態で、通信が行われていません。参照のみ可能です。

■構文

Object.Socket

Socket プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	UDPOCX オブジェクトです。

■データ型

長整数(Long)

■単語解説

ソケット

ソケットとはプロセス間通信によるデータ送受信を行うアプリケーションを作成するためのプログラミング・インターフェース(API)です。

State プロパティ

■機能

UDPOCX の状態を示します。参照のみ可能です。

■構文

Object.State

State プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	UDPOCX オブジェクトです。

■参照値

参照値は次の通りです。

(指定項目)		(内容)
STATE_CLOSED	0	接続していません
STATE_OPEN	1	開いています
STATE_LISTENING	2	接続聴取中です
STATE_CONNECTION_PENDING	3	接続待機中です
STATE_RESOLVING_HOST	4	ホスト名解決中です
STATE_HOST_RESOLVED	5	ホスト名が解決しました
STATE_CONNECTIONING	6	接続処理中です
STATE_CONNECTED	7	接続しました
STATE_CLOSING	8	切断処理中です
STATE_ERROR	9	エラーです

■データ型

整数(Integer)

UserData1～5 プロパティ

■機 能

アプリケーションが自由に利用できるデータ領域です。バイナリデータは格納できません。

■構 文

Object.UserData[=Value]

UserData プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
Value	データ領域を指定する文字列式です。

■データ型

文字列(String)

UserFlag プロパティ

■機 能

アプリケーションが自由に利用できるデータ領域です。バイナリデータは格納できません。

■構 文

Object.UserFlag[=Value]

UserFlag プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
Value	データ領域を指定する文字列式です。

■データ型

文字列(String)

WinsockVer プロパティ

■機 能

UDPOCX が使用している Winsock のバージョンを格納しています。参照のみ可能です。

■構 文

Object.WinsockVer

WinsockVer プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

WinsockDesc プロパティ

■機 能

UDPOCX が使用している Winsock の情報を格納しています。参照のみ可能です。

■構 文

Object.WinsockDesc

WinsockDesc プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

WinsockSysStat プロパティ

■機 能

UDPOCXが使用しているWinsockのシステム状況を格納しています。参照のみ可能です。

■構 文

Object.WinsockSysStat

WinsockSysStat プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

WinsockMaxSockets プロパティ

■機 能

UDPOCX が使用している Winsock が生成できるソケットの最大数を格納しています。参照のみ可能です。

■構 文

Object.WinsockMaxSockets

WinsockMaxSockets プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

WinsockMaxDatagram プロパティ

■機 能

UDPOCX が使用している Winsock が生成できる datagram の最大パケット数を格納しています。参照のみ可能です。

■構 文

Object.WinsockMaxDatagram

WinsockMaxDatagram プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

■データ型

文字列(String)

メソッド

メソッド

白紙ページ

Open メソッド

■機 能

Open メソッドはソケットを作成します。ポート番号が指定された場合は、そのポートに bind します。通常、明示的にポートを指定すると、そのソケットはサーバーとして振舞うことができます。ただし、“0”が指定された場合は、Winsock により自動的に空きポートが割り振られます。また、ポート番号が指定されていない場合、UDPOCX は“0”が指定されたものとして処理します。この場合、このソケットは主にクライアントとして振舞います。

■構 文

Object.Open(LocalPort As String)

Open メソッドの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
LocalPort	ローカルのポート番号、またはサービス名を指定します。空文字列も指定可能です。

■戻り値

なし。

■解 説

クライアントアプリケーションを作成する場合は、ポート番号に空文字列を指定してください。

SendTo メソッド

■機 能

データを送信します。

■構 文

Object. SendTo (RemoteIp As String, RemotePort As String, Data As String)

SendTo メソッドの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
RemoteIp	送信相手の IP アドレスを指定します。ホスト名の指定も可能です。
RemotePort	送信相手のポート番号を指定します。サービス名の指定も可能です。
Data	送信するデータを指定します。

■戻り値

なし。

Close メソッド

■機能

通信を終了します。

■構文

Object.Close()

Close メソッドの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	UDPOCX オブジェクトです。

■戻り値

なし。

■解説

このメソッドを呼び出すと、送信待ちのデータはすべて破棄されます。最終データの送信後、通信を終了する場合は、送信完了の Sent イベントを受け取ってから Close メソッドを使用してください。Sent イベントを受け取らずに Close メソッドを使用するとデータが送信されずに通信を終了することがあります。

htonl メソッド

■機 能

ホストバイトオーダーのバイナリ値をネットワークバイトオーダーに変換します。戻り値 net は String 型ですが、先頭から 4 バイト分に有効な値が入っています。

■構 文

```
net = Object htonl(host As long)
```

htonl メソッドの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
host	ホストバイトオーダーの長整数(4 バイト)値です。
net	ネットワークバイトオーダーの値です。型は String ですが、実際にはバイナリ値が格納されています。

■戻り値

文字列(String)

■解 説

Send メソッドを使用してバイナリ値を送る場合、バイトオーダーが重要なデータの場合、このメソッドで変換してください。net は必ず変数を用いてください。変数以外を指定するとエラーになります。

htons メソッド

■機能

ホストバイトオーダーのバイナリ値をネットワークバイトオーダーに変換します。戻り値 net は String 型ですが、先頭から 2 バイト分に有効な値が入っています。

■構文

```
net = Object.htons(host As Integer)
```

htons メソッドの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	UDPOCX オブジェクトです。
host	ホストバイトオーダーの整数(2 バイト)値です。
net	ネットワークバイトオーダーの値。型は String ですが、実際にはバイナリ値が格納されています。

■戻り値

文字列(String)

■解説

Send メソッドを使用してバイナリ値を送る場合、バイトオーダーが重要なデータの場合、このメソッドで変換してください。net は必ず変数を用いてください。変数以外を指定するとエラーになります。

ntohl メソッド

■機 能

ネットワークバイトオーダーのバイナリ値をホストバイトオーダーに変換します。引数 `net` は `String` 型ですが、このメソッドは先頭から 4 バイトだけを変換対象とします。

■構 文

```
host = Object.ntohl(net As String)
```

`ntohl` メソッドの構文の指定項目は次の通りです。

(指定項目)	(内 容)
<code>Object</code>	UDPOCX オブジェクトです。
<code>host</code>	ホストバイトオーダーの長整数(4 バイト)値です。
<code>net</code>	ネットワークバイトオーダーの値。型は <code>String</code> ですが、実際にはバイナリ値を指します。

■戻り値

長整数(Long)

■解 説

`Received` イベントの引数 `Data` にバイナリ値が含まれていてバイトオーダーが重要なデータのときは、このメソッドで変換してください。

ntohs メソッド

■機 能

ネットワークバイトオーダーのバイナリ値をホストバイトオーダーに変換します。引数 net は String 型ですが、このメソッドは先頭から 2 バイトだけを変換対象とします。

■構 文

host = Object.ntohs(net As String)

ntohs メソッドの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
host	ホストバイトオーダーの整数(2 バイト)値です。
net	ネットワークバイトオーダーの値。型は String ですが、実際にはバイナリ値を指します。

■戻り値

整数(Integer)

■解 説

Received イベントの引数 Data にバイナリ値が含まれていてバイトオーダーが重要なデータのときは、このメソッドで変換してください。

HostIp メソッド

■機 能

接続先に応じたローカルアドレスを得ます。

■構 文

LocalIp = HostIp(RemoteHost As String)

HostIp メソッドの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
LocalIp	ローカル IP アドレスです。
RemoteHost	接続先アドレスです。

■戻り値

文字列(String)

■解 説

通信で使用されるローカルアドレスは、接続先によって変わることがあります。

HostIp は、接続先(RemoteHost)へのコネクションを作成してローカルアドレスを得ます。Windows 2000 のみ動作します。

尚、Windows 2000 以外では、ホスト名に対応した IP アドレスを得ます。

イベント

イベント

白紙ページ

Ready イベント

■機 能

Open メソッドで作成したソケットが書き込み可能状態になったときに発生します。なお、ソケットにエラーが発生すると、一時的に書き込み不可能状態になることがあります。エラーが復旧し、再び書き込み可能状態になったときには、Ready イベントは発生しません。

■構 文

Private Sub Object_Ready()

Ready イベントの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

Received イベント

■機 能

通信相手からデータが到着したときに発生します。渡される引数に受け取ったデータが格納されています。

■構 文

Private Sub Object_Received(ByVal IPAddress As String, ByVal Port As String, ByVal Data As String)

Received イベントの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
IPAddress	通信相手の IPAddress (ドット区切り形式)が格納されます。
Port	通信相手のポート番号の整数の文字列式が格納されます。
Data	socket から読み込んだデータの文字列式が格納されます。

■解 説

渡されたデータを Visualbasic の文字列処理関数で処理するには、Unicode に変換しなくてはなりません。

Sent イベント

■機 能

Sendメソッドが実行された後、再び送信可能状態になったとき発生します。このとき、Sendable は True となっています。

■構 文

Private Sub Object_Sent()

Sent イベントの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。

WsError イベント

■機 能

何らかのエラーが発生したときに発生します。この後、Closed イベントが発生します。なお、CancelClose は UDPOCX の内部デバッグ用途にのみ使用しますので、設定しないでください。

■構 文

```
Private Sub Object_WsError(ByVal Ecode As Long, ByVal Description As String, ByVal Where As String, CancelClose As Boolean)
```

WsError イベントの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	UDPOCX オブジェクトです。
Ecode	エラーコードです。
Description	エラー内容です。
Where	エラーが発生した関数名です。
CancelClose	このイベント後に Close 処理をするかどうか指定します。次の「設定値」を参照してください。

■設定値

CancelClose の設定値は次の通りです。

(値)	(説 明)
True	Close 処理を行わない。
False	Close 処理を行う。(デフォルト)

Winsock エラーコード

Winsock エラーコード

白紙ページ

10004	システムコールの割り込みが発生
10009	無効なソケット番号を指定した
10013	アクセスが拒否された
10014	間違ったアドレスを指定した
10022	無効な引数
10024	開いているファイル数が多すぎる
10035	ブロッキングモードなら操作がブロックした
10036	ブロッキング処理中に別の API を呼び出した
10037	非ブロッキング処理中に同じ API を呼び出した
10038	ソケット以外に対してソケット操作が行われた
10039	宛先のアドレスが必要
10040	メッセージが長すぎる
10041	ソケットに対するプロトコルの種類が不正
10042	プロトコルが使用できない
10043	プロトコルがサポートされていない
10044	ソケットの種類がサポートされていない
10045	操作がソケットでサポートされていない
10046	プロトコルファミリがサポートされていない
10047	アドレスファミリがサポートされていない
10048	アドレスがすでに使用中
10049	要求されたアドレスを割り当てられない
10050	ネットワークが落ちている
10051	ネットワークが到達できない
10052	リセットによりネットワーク接続が落とされた
10053	ホスト内部で接続を中断した
10054	相手が接続を強制的に切った
10055	利用可能なバッファスペースが存在しない
10056	ソケットがすでに接続されている
10057	ソケットが接続されていない
10058	ソケットの遮断後であるため送信できない
10059	参照数が過大:結合不可能
10060	時間切れで connect または send に失敗した
10061	接続が拒否された
10062	シンボリックリンクが多すぎる
10063	ファイル名が長すぎる
10064	ホストが落ちている
10065	ホストへの到達経路が存在しない
10091	ネットワークサブシステムが使用不能
10092	Winsock がサポートしていないバージョンを指示した
10093	Winsock が初期化されていない
10101	接続が切れている
11001	ホストが見つからない
11002	やり直し可能なエラー
11003	回復不可能なエラー
11004	データがない

Winsock エラーコード

白紙ページ

サンプル

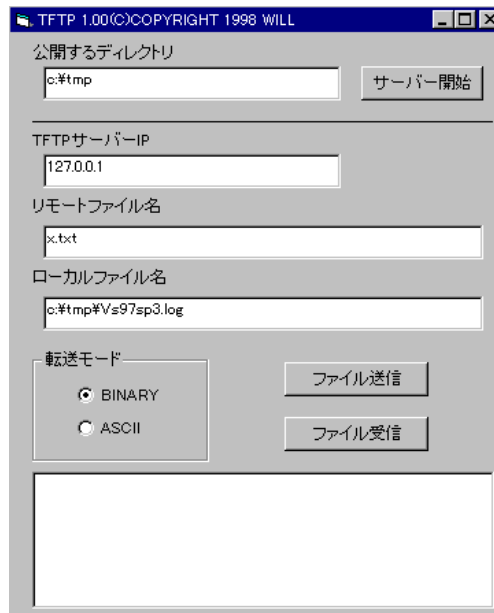
サンプル

白紙ページ

TFTP

(TFTP Ver1.0)

TFTP は、UDPOCX を用いた TFTP サーバーのサンプルプログラムです。TFTP とは、単純なファイル転送プロトコルです。異なるネットワークにあるマシン間のファイル転送を UDP で行っています。TFTP サンプルプログラムは、サーバーに接続するユーザーに開放するディレクトリを指定することができます。



■使い方

<サーバー機能>

- 公開するディレクトリ

公開するディレクトリを指定して下さい。「サーバー開始」ボタンを押すと指定したディレクトリが公開されます。

<クライアント機能>

- TFTP サーバーIP

公開するディレクトリのある(サーバーにする)マシンの IP アドレスを指定して下さい。

- リモートファイル名

公開するディレクトリにある(または新規に作成する場合の)ファイル名を指定して下さい。

- ローカルファイル名

ローカルマシンにある(または新規に作成する場合の)ファイル名を指定して下さい。

- 転送モード

ファイルの転送モードを選択して下さい。

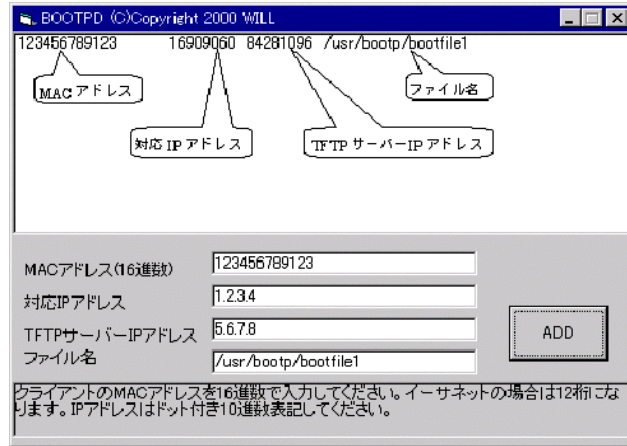
Bootpd

(Bootpd Ver1.0)

Bootpd は、ディスクレス・クライアント・マシンに自分自身の IP アドレスや、サーバーのアドレス、ロードして実行すべきファイルの名前を見つけて提供します。

その為に予め、各クライアントマシンの MAC アドレスに対して情報を設定しておきます。

以下に示す設定項目を入力後、「ADD」ボタンを押すと、リストボックスに提供する情報が追加されます。



■使い方

- **MAC アドレス (16 進数)**

クライアントのマシンに予め指定されている MAC アドレスを 16 進数で入力します。イーサネットの場合は 12 桁になります。

- **対応 IP アドレス**

ドット付きの 10 進数で入力して下さい。リストボックスには、32 ビット整数表示されません。

- **TFTP サーバーIP アドレス**

使用する TFTP サーバーの IP アドレスを指定して下さい。

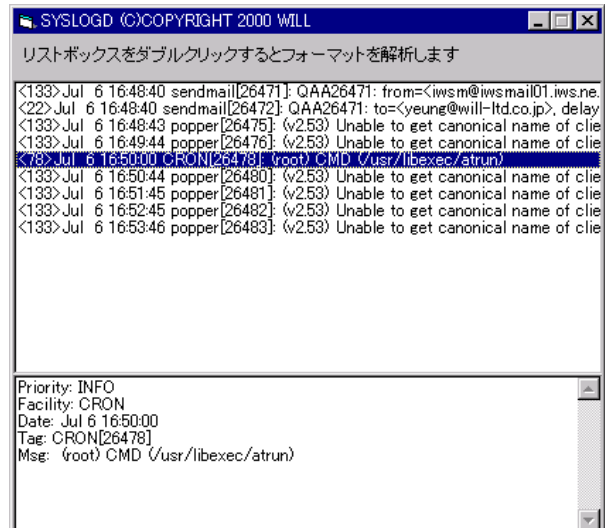
- **ファイル名**

TFTP を行うファイル名を指定して下さい。

Syslogd

(Syslogd Ver1.0)

Syslogd は、UNIX の Syslog サーバーからの転送データを表示します。転送されてきたデータをクリックすると、情報が表示されます。



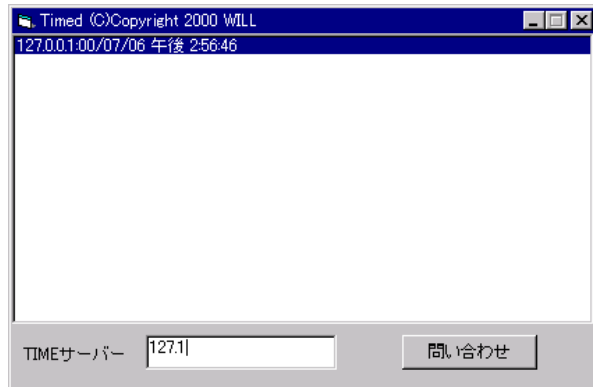
The screenshot shows a window titled "SYSLOGD (C)COPYRIGHT 2000 WILL". The main area contains a list of log messages. One message is highlighted in blue: "<133>Jul 6 16:50:00 CRON[26478]: (root) CMD (/usr/libexec/atrun)". Below the list, a detailed view of the selected message is shown:

```
Priority: INFO
Facility: CRON
Date: Jul 6 16:50:00
Tag: CRON[26478]
Msg: (root) CMD (/usr/libexec/atrun)
```

timed

(timed Ver1.0)

ブロードキャストでサーバーに要求できる UDP を使って、サーバーに時刻を問い合わせして表示します。



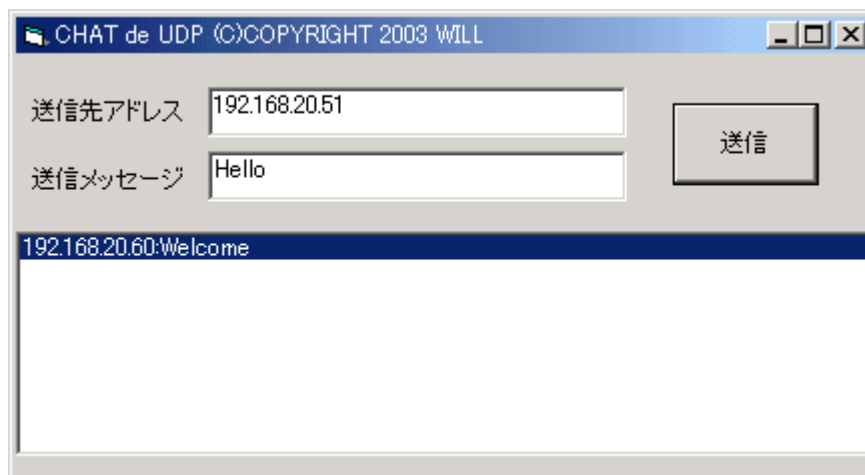
■使い方

- TIME サーバーの IP アドレスを指定して、「問い合わせ」ボタンを押して下さい。リストボックスに時刻が表示されます。
- ブロードキャストで問い合わせる場合は、IP アドレスを「255.255.255.255」と指定して下さい。

CHAT de UDP

(CHAT de UDP Ver1.00)

UDP を利用したチャットサンプルです。ブロードキャストの IP を指定すると1対多でのチャットが行えます。



■使い方

1. 送信先アドレスのテキストボックスに、IP アドレスまたは、ホスト名を指定します。このとき、ブロードキャストの IP を指定すると、起動している「CHAT de UDP」全てにメッセージを送ることができます。
2. 送信メッセージのテキストボックスに、メッセージを入力します。
3. 「送信」ボタンを押すと、メッセージが送信されます。
4. 下段のリストボックスに受信したメッセージが表示されます。

※ポート番号は、50001 固定です。

サンプル

白紙ページ

索引

AnyData プロパティ.....	27
Broadcast プロパティ.....	28
Close メソッド.....	49
Copyright プロパティ.....	29
HostIP メソッド.....	54
htonl メソッド.....	50
htons メソッド.....	51
LastError プロパティ.....	30
LocalAddress プロパティ.....	32
LocalHost プロパティ.....	31
LocalIp プロパティ.....	33
LocalPort プロパティ.....	34
ntohl メソッド.....	52
ntohs メソッド.....	53
Open メソッド.....	47
Ready イベント.....	57
Received イベント.....	58
Sendable プロパティ.....	35
SendTo メソッド.....	48
Sent イベント.....	59
Socket プロパティ.....	36
State プロパティ.....	37
UserData1~5 プロパティ.....	38
UserFlag プロパティ.....	39
WinsockDesc プロパティ.....	41
WinsockMaxDatagram プロパティ.....	44
WinsockMaxSockets プロパティ.....	43
WinsockSysStat プロパティ.....	42
WinsockVer プロパティ.....	40
WsError イベント.....	60

白紙ページ

UDPOCX マニュアル

1999年1月14日 初版第1版

2001年5月15日 第2版

2002年9月1日 第3版

発行所 株式会社ウィル

住所 神奈川県横浜市保土ヶ谷区西久保町15

グランディシヤ 302

〒240-0022

TEL: 045-338-3525

FAX: 045-338-3526

Mail-Address: info@will-ltd.co.jp

URL: <http://www.will-ltd.co.jp/>

発行者 小川 史彦

本紙の内容を許可なく複写、転載、データファイル化することを禁じます。
本紙の内容に関するご質問は、上記のメールアドレス宛にお問い合わせください。