

# SNMPOCX

WILL

株式会社ウィル

白紙ページ

白紙ページ

- Microsoft、Windows、Windows NT、Windows XP、 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
Ver 1.XX からの移行について.....	15
再配布について.....	16
製品概要.....	19
特徴.....	21
SNMPSEND コントロール.....	21
SNMPTRAP コントロール.....	21
SNMPSEND コントロール.....	23
▼ プログラミング概要.....	25
▼ プロパティ.....	29
Stat プロパティ.....	31
AgentAddress プロパティ.....	33
TickCount プロパティ.....	34
StringType プロパティ.....	35
▼ メソッド.....	37
Err2Str メソッド.....	39
ClearOID メソッド.....	40
SetOID メソッド.....	41
OID2Name メソッド.....	42
Name2OID メソッド.....	43
FindName メソッド.....	44
Name2TypeID メソッド.....	45
SNMPNullValue メソッド.....	46
SNMPCounterValue メソッド.....	47
SNMPIntegerValue メソッド.....	48
SNMPStringValue メソッド.....	49
SNMPIpAddValue メソッド.....	50

SNMPGaugeValue メソッド .....	51
SNMPTimeTicksValue メソッド .....	52
SNMPBinStringValue メソッド .....	53
SendGetRequest メソッド .....	54
SendGetNextRequest メソッド .....	55
SendSetRequest メソッド .....	56
SendTrap メソッド .....	57
▼ イベント .....	59
Response イベント .....	61
Ready イベント .....	63
WsError イベント .....	64
SNMPTRAP コントロール .....	65
▼ プログラミング概要 .....	67
▼ プロパティ .....	71
Stat プロパティ .....	73
StringType プロパティ .....	74
▼ メソッド .....	75
StartTrap メソッド .....	77
StopTrap メソッド .....	78
▼ イベント .....	79
Ready イベント .....	81
Trap イベント .....	82
WsError イベント .....	84
サンプル .....	85
SNMP Manager サンプル (SNMP MANAGER 2.01) .....	87
SENDTRAP サンプル .....	89
索引 .....	91

**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  
(WILLWARE Components セットアップキット)
  - Cryptdll.exe  
(暗号 DLL 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

SNMPOCX 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

### ■開発に必要な環境

SNMPOCX 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)

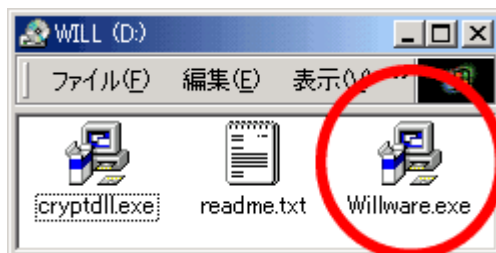
尚 SNMPOCX を動作させるには、以下の環境が必要ですが、開発環境ではインストールの際に自動的に組み込まれます。

VB5 ランタイム DLL(SP3 以上)  
MFC42 ランタイム DLL  
UDPOCX 1.07  
INT64DLL 1.01

SNMPOCX は、Microsoft Visual Basic Ver 5.0 で作成しています。サンプルは、Microsoft Visual Basic Ver 5.0 及び Ver 4.0 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。

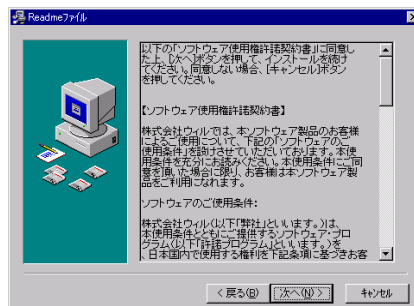


画面にしたがって、インストールを進めて下さい。

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。



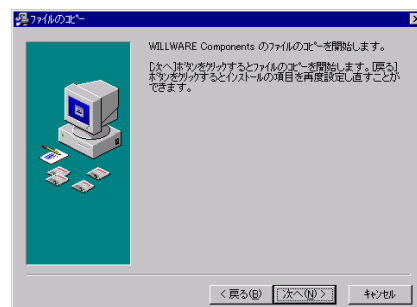
4. インストール中に置換されるファイルのバックアップを作成できます。そのバックアップファイルの保存先フォルダを指定します。初期設定でよろしければ「次へ」をクリックして下さい。



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 (EPXXXXXXXXX.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。)

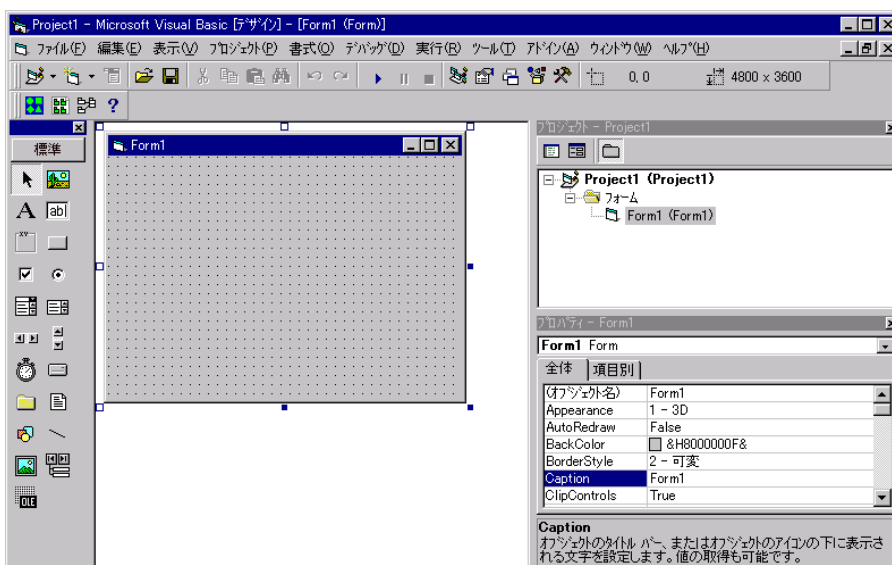


以下のメッセージボックスが表示され、ライセンスがレジストリに登録されます。



## ■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「WILL SNMP Send OCX」及び「WILL SNMP Trap OCX」を選択して「OK」をクリックすると、それぞれのアイコンがツールボックスに追加されます。



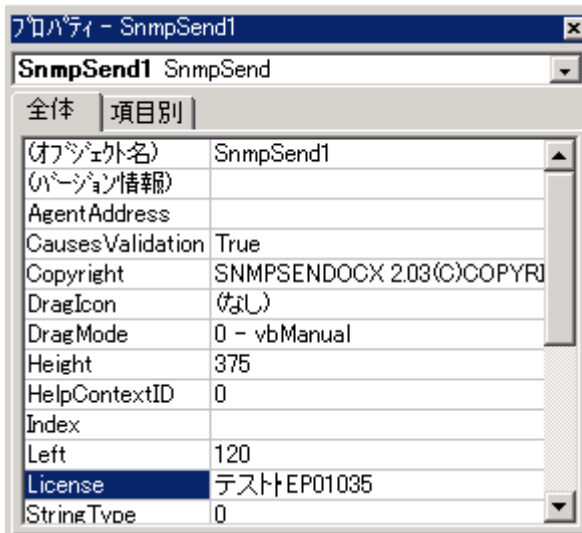
ツールボックスに追加された OCX を選択し、フォームにアイコンを貼り付け、プロパティの「(バージョン情報)」をクリックすると、以下の「WILL LICENSE REGISTRATION」画面が表示されます。ここで、ユーザー名、シリアル番号、キーコードをそれぞれ入力してライセンスの登録を行います。



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 OCX 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、SNMPOCX 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!?

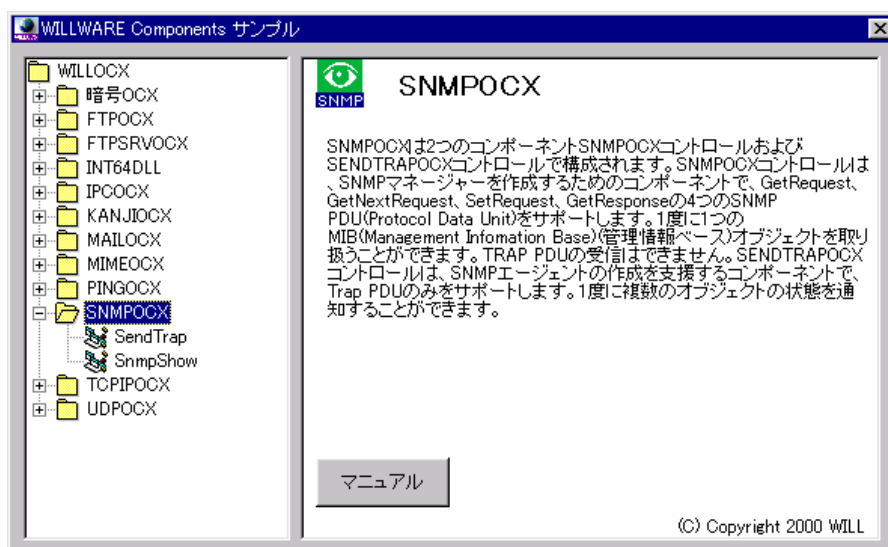
既にトライアルライセンスで作成したアプリケーションは、正規ライセンスを登録した後、再コンパイルする必要があります。

## 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. サーバーの種類
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。

## ■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

はじめに

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。

# Ver 1.XX からの移行について

SNMPOCX を大幅に変更し Ver 2.00 にバージョンアップいたしました。

主な変更点は、

- 複数の MIB オブジェクトを一度に取り扱うことができるようにしました。
- TRAP 受信を可能にしました。

です。

尚、バージョンアップではありますが、従来のコントロールとはバイナリ互換がありませんので、別コントロールとしてご提供します。従来のソースコードから移行される場合は、下記の注意すべきメソッドとイベントを参考にして、ソースコードの変更をお願いします。

## ■ コントロールの対応

(従 来)	(Ver 2.00)
SNMP.OCX	SNMPSEND.OCX (SNMP SendOCX 2.00)
SENDTRAP.OCX	SNMPSEND.OCX に統合
-----	<新規>SNMPTRAP.OCX (SNMP TrapOCX 2.00)

## ■ メソッドの名称の変更

(従 来)	(Ver 2.00)
GetSNMPString メソッド	SNMPStringValue メソッド
GetSNMPInteger メソッド	SNMPIntegerValue メソッド
GetSNMPCounter メソッド	SNMPCounterValue メソッド

## ■ メソッドの名称及び引数の変更

(従 来)	(Ver 2.00)
GetRequest メソッド	SendGetRequest メソッド
GetNextRequest メソッド	SendGetNextRequest メソッド
SetRequest メソッド	SendSetRequest メソッド

## ■ イベントの引数の変更

Response イベント

## 再配布について

---

### ■作成したアプリケーションの配布時

SNMPOCX 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

SNMPOCX 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。

SNMPSEND.OCX(2.00)
SNMPTRAP.OCX(2.00)
UDP.OCX(1.07)
INT64.DLL(1.01)
VB5JP.DLL (Ver 5.0.4319)
MSVBVM50.DLL (Ver 5.2.8244)
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

SNMPOCX 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 (Windows 95, 98 の場合)

C:¥WINNT¥system32 (Windows NT, 2000, 2003 の場合)

C:¥Windows¥system32 (Windows XP 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

- ・ SNMPOCX 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。
- ・ SNMPOCX を利用するアプリケーションは SNMPOCX 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 SNMPOCX 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、SNMPOCX を使用していることを示してください。

はじめに

# 製品概要

製品概要

白紙ページ



## 特 徴

---

SNMPOCX は、SNMP V1 プロトコルをサポートする ActiveX コンポーネント群です。SNMP エージェント内にある MIB I 及び MIB II 或いは、プライベート MIB の値を表示・変更することができます。

SNMPOCX は 2 つのコンポーネント SNMPSEND コントロールおよび SNMPTRAP コントロールで構成されます。これら 2 つの総称を SNMPOCX と呼びます。

SNMPSEND コントロールは、SNMP マネージャーを作成するためのコンポーネントで、GetRequest、GetNextRequest、SetRequest、GetResponse の 4 つの SNMP PDU(Protocol Data Unit)をサポートします。1 度に複数の MIB(Management Information Base)(管理情報ベース)オブジェクトを取り扱うことができます。さらに、TRAP PDU の送信を行うことができます。

SNMPTRAP コントロールは、Trap PDU の受信をサポートするコンポーネントです。詳しくはそれぞれのマニュアルをご参照ください。

## SNMPSEND コントロール

---

SNMP はさまざまなネットワーク機器でサポートされるプロトコルです。SNMP を利用すればネットワーク機器の状態監視を簡単に行なえます。高機能なネットワーク機器は、機能変更を SNMP を利用して行なうことができます。SNMP をサポートするネットワーク機器を SNMP エージェントと呼びます。SNMPSEND コントロールは、VisualBasic など Active X Control を利用できる環境において SNMP エージェントから情報を収集または SNMP エージェントのコントロールを行なうことを可能とするものです。SNMPSEND コントロールのサンプルを試してみてください。ネットワーク機器の状態が簡単に入手できることに気がつくことでしょう。

## SNMPTRAP コントロール

---

SNMPTRAP コントロールは、VisualBasic など Active X Control を利用できる環境から、SNMP TRAP PDU を受信することを可能にするものです。

TRAP は、エージェントからマネージャへのイベント通知です。イベントには、エージェントの開始、通信リンク障害、認証エラー、ノードダウン等の既定のイベントの他、ユーザー定義イベントがあります。

製品概要

白紙ページ

# SNMPSEND コントロール

▼ プログラミング概要

▼ プロパティ

▼ メソッド

▼ イベント

白紙ページ

## ▼ プログラミング概要

白紙ページ

SNMPSEND コントロールの使い方は簡単です。

SNMP エージェントから情報を収集するためには、

SendGetRequest メソッド

SendGetNextRequest メソッド

の 2 つのメソッドを利用します。

収集したい情報には ID がついています。これをオブジェクト識別子(OID)と呼びます。OID がわかっている場合は、SendGetRequest メソッドを使います。

OID がわからない場合は、SendGetNextRequest を使って代表的な OID “1.3.6”の次に存在する OID を得ます。SendGetNextRequest を繰り返し使うことで SNMP エージェントに存在するすべての OID を取得することができます。(一部のエージェントはこのような使い方を禁止している場合があります)

SNMP エージェントに値を設定するには、

SendSetRequest メソッド

を使います。

たとえば、

iso.org.dod.internet.mgmt.mib.system.sysContact.0

や

iso.org.dod.internet.mgmt.mib.interfaces.ifTable.ifEntry.ifAdminStatus.1

を検索してみてください。これらの値は変更可能かもしれません。

他の SNMP マネージャーに通知を行うには SendTrap メソッドを使います。

SNMPSEND コントロールは、オブジェクト名をサポートするいくつかのメソッドがあります。OID をオブジェクト名で表現するとかなりわかりやすくなります。

OID をオブジェクト名で表現するには、

OID2Name メソッド

を使います。

オブジェクト名にどのようなものがあるのか検索するには、

FindName メソッド

を使います。このメソッドにたとえば、“sys”という文字列を与えると sys を含むオブジェクト名のリストを Collection にしてくれます。

OID とオブジェクト名の関係は

ClearOID メソッド

SetOID メソッド

で変更できます。これにより新規の OID に名前をつけることができます。

## 【用 語】

### ■ オブジェクト識別子(OID)

いくつかのピリオドで区切られた整数

(例: "1.3.6.1.2.1.1.2")

### ■ オブジェクト名

オブジェクト識別子に対する名前

(例: "iso.org.dod.internet.mgmt.mib.system.sysObjectID")

### ■ オブジェクトインスタンス

OID にさらに "." と番号を付加したもの。

### ■ スカラ値を持つものは ".0"、

配列値は ".1" から順番に番号を付加したものになる。

(例: "1.3.6.1.2.1.1.2.0")

(例: "iso.org.dod.internet.mgmt.mib.system.sysObjectID.0")



## ▼ プロパティ

白紙ページ

# Stat プロパティ

---

## ■機能

SNMPSEND コントロールの状態を示します。

## ■構文

Object.Stat

Stat 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	SNMPSEND コントロールオブジェクトです。

## ■データ型

整数(Integer)。

## ■戻り値

(定数)	(説明)
0	初期状態。送受信不可です。
1	送信可能状態です。
2	送信中です。送信不可です。

## ■解説

1つのサブルーチンから連続して送信を行なう場合、Stat プロパティを監視して送信を行ないます。

## ■Stat の使用例

```

WaitReady()
  Do While Snmp1.Stat <> 1
    DoEvents
  Loop
End Sub

SendMult()
  WaitReady
  Request1

```

## SNMPSEND コントロール

```
    WaitReady  
    Request2  
    WaitReady  
    Request3  
    WaitReady  
End Sub
```

# AgentAddress プロパティ

---

## ■機能

TRAP データに格納するエージェントアドレスを指定します。デフォルトは「'''」です。「'''」の場合は、データが送信されるときに利用されたネットワークカードに設定されている IP アドレスです。

このプロパティは、SendTrap メソッド内で参照します。

## ■構文

Object.AgentAddress[=Address]

AgentAddress 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	SNMPSEND コントロールオブジェクトです。
Address	ドット付 10 進数です。例: "192.173.56.2"。

## ■データ型

文字列(String)。

## ■設定値

設定した値です。

## ■解説

TRAP の原因となった機器の IP アドレスがわかる場合にのみ指定します。

「'''」を指定すれば PC の IP アドレスが設定されます。ホスト名を指定することはできません。

# TickCount プロパティ

---

## ■機 能

TRAP データに格納する TickCount(1/100 秒)を指定します。  
デフォルトは「-1」です。「-1」の場合は、データが送信されるときに PC の起動開始時からの経過時間を設定します。  
このプロパティは、SendTrap メソッド内で参照します。

## ■構 文

Object.TickCount [=Count]

TickCount 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	SNMPSEND コントロールオブジェクトです。
Count	経過時間(1/100 秒単位)です。

## ■データ型

長整数(Long)。

## ■設定値

設定した値です。

## ■解 説

TRAP の原因となった機器の TickCount がわかる場合にのみ利用します。  
「-1」を指定すれば PC の経過時間が設定されます。

# StringType プロパティ

---

## ■機能

受信した OctedString の取り扱いを指示します。

## ■構文

Object.StringType [=Value]

StringType 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	SNMPSEND コントロールオブジェクトです。
Value	文字列を Unicode に変換するかどうかを指示する値です。

## ■データ型

長整数(Long)。

## ■設定値

Value の設定値は次の通りです。

(定数)	(説明)
0	初期状態。Unicode に変換します。
1	Unicode に変換しません。





## ▼ メソッド

白紙ページ

# Err2Str メソッド

---

## ■機 能

ErrorStatus 変数に対するエラーメッセージを取得します。

## ■構 文

Object.Err2Str(ErrorStatus As Integer)

Err2Str 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	SNMPSEND コントロールオブジェクトです。
ErrorStatus	エラーメッセージを示す整数です。

## ■戻り値

文字列(String)。

## ■引数

ErrorStatus(引数)	(戻 値)
-1	"FormatError"
0	"NoError"
1	"TooBig"
2	"NoSuchName"
3	"BadValue"
4	"ReadOnly"
5	"GenError"
Else	"UnknownError"

# ClearOID メソッド

---

## ■機 能

SNMPSEND コントロールが内部に持っている OID とオブジェクト名リストをクリアします。

## ■構 文

Object.ClearOID()

ClearOID 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	SNMPSEND コントロールオブジェクトです。

## ■戻り値

なし

# SetOID メソッド

---

## ■機 能

SNMPSEND コントロールが内部に持っている OID と OID に対応する名前リストにデータを追加します。

## ■構 文

Object.SetOID(OID As String, Name As String)

SetOID 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	SNMPSEND コントロールオブジェクトです。
OID	オブジェクト識別子です。
Name	オブジェクト名です。

## ■戻り値

なし

## ■解 説

ここで設定した値は、OID2Name や name2OID、FindName 各メソッドで利用されます。OID を独自表記したい場合は、ClearOID をした後、SetOID を行なってください。

# OID2Name メソッド

---

## ■機 能

OID から OID に対応するオブジェクト名に変換します。

## ■構 文

Object.OID2Name(OID As String)

OID2Name 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	SNMPSEND コントロールオブジェクトです。
OID	オブジェクト識別子です。

## ■戻り値

文字列(String)。

## ■解 説

OID の一部または全部をオブジェクト名に変換します。

たとえば、“1.3.6.1.2.1.1.2.0”であれば、内部に、

“1.3.6.1.2.1.1.2”は“iso.org.dod.internet.mgmt.mib.system.sysObjectID”

であるという表がある場合には、

“iso.org.dod.internet.mgmt.mib.system.sysObjectID.0”

のように変換されます。

この表は、SetOID で変更可能です。

# Name2OID メソッド

---

## ■機 能

OID に対応するオブジェクト名を OID に変換します。

## ■構 文

Object.Name2OID (Name As String)

Name2OID 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	SNMPSEND コントロールオブジェクトです。
Name	オブジェクト名です。

## ■戻り値

文字列(String)。

## ■解 説

オブジェクト名を OID に変換します。

オブジェクト名と OID の対応が表に存在しなければなりません。

# FindName メソッド

---

## ■機 能

オブジェクト名を検索します。

## ■構 文

Object.FineName (Name As String, [Compere As Integer = 0])

FindName 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	SNMPSEND コントロールオブジェクトです。
Name	オブジェクト名の一部です。
Compere	0 = バイナリ比較です。 1 = テキスト比較です(大文字と小文字を区別しない)。

## ■戻り値

コレクション(Collection)。

## ■解 説

指定したオブジェクト名の一部が一致するオブジェクト名一覧を Collection として戻します。これを利用するとオブジェクト名の一部がわかっている場合に正式名称を得ることができます。



# Name2TypeID メソッド

---

## ■機能

OID に対応するオブジェクト名を OID に変換します。

## ■構文

Object.Name2TypeID(Name As String)

Name2TypeID 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	SNMPSEND コントロールオブジェクトです。
Name	オブジェクト名です。

## ■戻り値

整数(Integer)。

## ■引数

Name(引数)	(戻 値)
"Integer"	2
"OctetString"	4
"Null"	5
"ObjectID"	6
"Sequence"	&H30
"IpAddress"	&H40
"Counter"	&H41
"Gauge"	&H42
"Timeticks"	&H43
"Opaque"	&H44
"GetRequest"	&HA0
"GetNextRequest"	&HA1
"GetResponse"	&HA2
"SetRequest"	&HA3
"Trap"	&HA4
Else	-1

# SNMPNullValue メソッド

---

## ■機 能

NULL 値を持つオブジェクト ID を示すデータを生成します。

## ■構 文

Object.SNMPNullValue(ObjectId As String)

SNMPNullValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。

## ■戻り値

文字列(String)。

## ■解 説

SendGetRequest や、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。

生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

# SNMPCounterValue メソッド

---

## ■機能

符号なし整数型のオブジェクト ID とその値を示すデータを生成します。

## ■構文

Object.SNMPCounterValue(ObjectId As String, Value As String)

SNMPCounterValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。
Value	整数です。例: "12345"。

## ■戻り値

文字列(String)。

## ■解説

ObjectValue には、32 ビットまでの整数値を示す文字列を設定できます。

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。

生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

# SNMPIntegerValue メソッド

---

## ■機能

文字列型のオブジェクト ID とその値を示すデータを生成します。

## ■構文

Object.SNMPIntegerValue(ObjectId As String, Value As String)

SNMPIntegerValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。
Value	整数です。例: "12345"。

## ■戻り値

文字列(String)。

## ■解説

ObjectValue には、32 ビットまでの整数値を示す文字列を設定できます。

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。

生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

# SNMPStringValue メソッド

---

## ■機能

文字列型のオブジェクト ID とその値を示すデータを生成します。

## ■構文

Object.SNMPStringValue(ObjectId As String, Value As String)

SNMPStringValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。
Value	文字列です。例: "sample"。

## ■戻り値

文字列(String)。

## ■解説

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。

生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

# SNMPIpAddressValue メソッド

---

## ■機 能

IP アドレス型のオブジェクト ID とその値を示すデータを生成します。

## ■構 文

Object.SNMPIpAddressValue(ObjectId As String, Value As String)

SNMPIpAddressValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。
Value	IP アドレスです。例: "192.58.6.4"。

## ■戻り値

文字列(String)。

## ■解 説

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。

生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

# SNMPGaugeValue メソッド

---

## ■機能

Gauge 型のオブジェクト ID とその値を示すデータを生成します。

## ■構文

Object. SNMPGaugeValue(ObjectId As String, Value As String)

SNMPGaugeValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。
Value	整数です。

## ■戻り値

文字列(String)。

## ■解説

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。

生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

## ■参考

Gauge 型とは、上限が 4、294、967、295 である 0 またはそれ以上の整数です。

# SNMPTimeTicksValue メソッド

---

## ■機 能

TimeTicks 型のオブジェクト ID とその値を示すデータを生成します。

## ■構 文

Object.SNMPTimeTicksValue(ObjectId As String, Value As String)

SNMPTimeTicksValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。
Value	経過時間を示す整数です。

## ■戻り値

文字列(String)。

## ■解 説

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。

生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

## ■参 考

TimeTicks 型とは、1/100 秒単位のある時刻からの経過時間です。



# SNMPBinStringValue メソッド

---

## ■機能

文字列型のオブジェクト ID とその値を示すデータを生成します。

## ■構文

Object.SNMPBinStringValue(ObjectId As String, Value As String)

SNMPBinStringValue 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	SNMPSEND コントロールオブジェクトです。
ObjectId	オブジェクト ID です。例: "1.3.6.1.4.1.5000.1"。
Value	バイナリ文字列です。例: ChrB\$(&HFF) & ChrB\$(&HFE)

## ■戻り値

文字列(String)。

## ■解説

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap の各メソッドのオブジェクトバリュー引数を、このメソッドを用いて作成します。生成されたデータは文字列型ですが、実態はバイナリですので、そのまま表示することはできません。

# SendGetRequest メソッド

## ■機能

エージェントからのオブジェクトインスタンスの値を取得します。  
Ready イベントが発生していることが必要です。送信後は、再度 Ready イベントの発生を待ってください。

## ■構文

Object.SendGetRequest

(RemoteIP As String, RemotePort As String, Community As String, RequestId As Long, ObjectValue As String)

SendGetRequest 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	SNMPSEND コントロールオブジェクトです。
RemoteIP	エージェントの IP アドレス。ホスト名でも結構です。
RemotePort	エージェントのポート番号。ポート名でも結構です。
Community	パスワードに相当します。
RequestId	任意の数字です。
ObjectValue	オブジェクトバリューです。

## ■戻り値

なし

## ■解説

GetRequest PDU メッセージを利用してエージェントにオブジェクトインスタンスの値の問い合わせを行いません。エージェントがこのメッセージを受信したら、GetResponse PDU メッセージを利用して応答します。

応答が無い場合には、RemoteIP,RemotePort,Community のいずれかが間違っている可能性があります。

また、RemoteIP や RemotePort に正しくない値をいれると例外が発生します。

Ready イベントが発生するか、Stat プロパティが 1 であることを確認してから、このメソッドを発行してください。

オブジェクトバリューは、SNMPNullValue の戻り値を文字列として結合したものです。

# SendGetNextRequest メソッド

## ■機能

エージェントからオブジェクトインスタンスの値を取得します。

Ready イベントが発生していることが必要です。送信後は、再度 Ready イベントの発生を待ってください。

## ■構文

Object.SendGetNextRequest

(RemoteIP As String, RemotePort As String, Community As String, RequestId As Long, ObjectValue As String)

SendGetNextRequest 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	SNMPSEND コントロールオブジェクトです。
RemoteIP	エージェントの IP アドレス。ホスト名でも結構です。
RemotePort	エージェントのポート番号。ポート名でも結構です。
Community	パスワードに相当します。
RequestId	任意の数字です。
ObjectValue	オブジェクトバリューです。

## ■戻り値

なし

## ■解説

GetNextRequest PDU メッセージを利用してエージェントに指定したオブジェクトインスタンスのつぎのオブジェクトインスタンス値の問い合わせを行いません。エージェントがこのメッセージを受信したら、GetResponse PDU メッセージを利用して応答します。

応答が無い場合には、RemoteIP,RemotePort,Community のいずれかが間違っている可能性があります。

また、RemoteIP や RemotePort に正しくない値をいれると例外が発生します。

Ready イベントが発生するか、Stat プロパティが 1 であることを確認してから、このメソッドを発行してください。

オブジェクトバリューは、SNMPNullValue の戻り値を文字列として結合したものです。

# SendSetRequest メソッド

## ■機 能

エージェントのオブジェクトインスタンスに値を設定します。  
Ready イベントが発生していることが必要です。送信後は、再度 Ready イベントの発生を待ってください。

## ■構 文

Object.SendSetRequest

(RemoteIP As String, RemotePort As String, Community As String, RequestId As Long, ObjectValue As String)

SendSetRequest 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	SNMPSEND コントロールオブジェクトです。
RemoteIP	エージェントの IP アドレス。ホスト名でも結構です。
RemotePort	エージェントのポート番号。ポート名でも結構です。
Community	パスワードに相当します。
RequestId	任意の数字です。
ObjectValue	設定するオブジェクトバリューです。

## ■戻り値

なし

## ■解 説

SetRequest PDU メッセージを利用してエージェントの指定したオブジェクトインスタンスに値を設定します。エージェントがこのメッセージを受信したら、GetResponse PDU メッセージを利用して応答します。値を設定できないオブジェクトインスタンスの場合は NoSuchName エラーが、正しくない値を設定しようとする BadValue エラーが Response イベントの ErrorStatus に格納されます。応答が無い場合には、RemoteIP, RemotePort, Community のいずれかが間違っている可能性があります。

また、RemoteIP や RemotePort に正しくない値をいれると例外が発生します。

Ready イベントが発生するか、Stat プロパティが 1 であることを確認してから、このメソッドを発行してください。

オブジェクトバリューは、SNMPNullValue, SNMPCounterValue, SNMPIntegerValue, SNMPStringValue, SNMPIpAddressValue, SNMPGaugeValue, SNMPTimeTicksValue の戻り値を文字列として結合したものです。

# SendTrap メソッド

## ■機能

TRAP データを送信します。

Ready イベントが発生していることが必要です。送信後は、再度 Ready イベントの発生を待ってください。

## ■構文

Object.SendTrap

(RemoteIP As String, RemotePort As String, Community As String, Enterprise As String, GenericTrapType As Integer, SpecificTrapType As String, [ObjectValue As String = ""]) )

SendTrap 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	SNMPSEND コントロールオブジェクトです。
RemoteIP	TRAP を受信する SNMP マネージャ(サーバ)です。
RemotePort	TRAP を受信する SNMP マネージャのポート番号です。ポート名でも結構です。
Community	コミュニティ名です。
Enterprise	TRAP を発信する企業を示す ObjectID です。 例: "1.3.6.1.4.1.5000"
GenericTrapType	TRAP タイプです。
SpecificTrapType	TRAP 詳細タイプです。
ObjectValue	トラップデータ(必要な場合のみ)です。

## ■戻り値

なし

## ■解説

Enterprise には、企業を示すオブジェクト ID を設定します。

GenericTrapType には、TRAP のタイプを指定します。

TRAP のタイプは 7 種類あります。

- 0 ColdStart
- 1 WarmStart
- 2 LinkDown
- 3 LinkUp
- 4 AuthenticationFailure

## SNMPSEND コントロール

5 EgpNeighborLoss

6 EnterpriseSpecific

SpecificTrapType には、任意の整数を設定できます。

オブジェクトバリューは、SNMPNullValue, SNMPCounterValue, SNMPIntegerValue, SNMPStringValue, SNMPIpAddressValue, SNMPGaugeValue, SNMPTimeTicksValue の戻り値を文字列として結合したものです。

## ▼ イベント

白紙ページ



# Response イベント

## ■機能

エージェントから応答があった場合に発生します。

## ■構文

Private Sub Object\_Response

(ByVal RemoteIP As String, ByVal RemotePort As String, ByVal Version As Integer, ByVal Community As String, ByVal Command As String, ByVal RequestId As Long, ByVal ErrorStatus As Integer, ObjectCollection As Collection, ByVal RawData As String)

Response 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	SNMPSEND コントロールオブジェクトです。
RemoteIP	エージェントの IP アドレスです。(ドット付き 10 進数)
RemotePort	エージェントのポート番号です。(通常 161)
Version	使用している SNMP バージョン番号です。(0:Version1)。固定値。
Community	コミュニティ名です。
Command	送信した PDU のタイプ(A2)です。
RequestId	指定したリクエスト ID です。
ErrorStatus	エラー番号。Err2Str メソッドを使って文字列にできます。
ObjectCollection	Collection 型のオブジェクトです。SNMPSendObject クラスが格納されています。
RawData	受信した生データです(バイナリ)。

## ■解説

RESPONSE PDU を受信したときに発生します。

ObjectCollection は、SnmSendObject の集合です。

ObjectCollection の内容を確認するには以下のようにコーディングします。

```
Dim v As SnmpSendObject
For Each v In ObjectCollection
    Debug.Print "Object ID=" & v.OIDName
    Debug.Print "Object TYPENAME=" & v.OIDTypeName
    Debug.Print "Object TYPE=" & v.OIDType
    Debug.Print "Object VALUE=" & v.OIDValue
Next
```

あるいは下記のようにコーディングします。

```
Dim i&
For i = 0 To ObjectCollection.Count - 1
    Debug.Print "Object ID=" & ObjectCollection (CStr(i)).OIDName
    Debug.Print "Object TYPENAME=" & ObjectCollection (CStr(i)).OIDTypeName
    Debug.Print "Object TYPE=" & ObjectCollection (CStr(i)).OIDType
    Debug.Print "Object VALUE=" & ObjectCollection (CStr(i)).OIDValue
Next
```

# Ready イベント

---

## ■機 能

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap を発行できる状態になったとき発生します。

## ■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	SNMPSEND コントロールオブジェクトです。

## ■解 説

SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap を連続して発行することはできません。

上記いずれかのメソッドを発行する場合は、Ready イベントを待ってください。

## ■関連事項

Stat プロパティ。

# WsError イベント

---

## ■機 能

SNMPSEND コントロール内で Winsock エラーを検出した場合に発生します。

## ■構 文

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

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	SNMPSEND コントロールオブジェクトです。
Ecode	Winsock エラーコードです。
Description	エラー番号に対応するエラーメッセージです。
Where	エラー発生場所です。

## ■解 説

このイベントが発生するとエラーが発生する前に送信した SNMP メッセージに対する応答はすべて破棄されます。処理を再開するには、エラー後に発生する Ready イベントを待つて再度 SendGetRequest、SendGetNextRequest、SendSetRequest、SendTrap を発行してください。問題解析のためにこれらの情報を記録してください。

# SNMPTRAP コントロール

▼ プログラミング概要

▼ プロパティ

▼ メソッド

▼ イベント

白紙ページ

## ▼ プログラミング概要

白紙ページ



SNMPTRAP コントロールは TRAP PDU を受信します。

受信を開始するために、StartTrap メソッドを使います。  
 受信を終了するには、StopTrap メソッドを使います。

TRAP PDU を受信すると、Trap イベントが発生します。

Trap イベントには、11 のパラメータがあります。

RemoteIP\$	Trap を発生したマシンの IP アドレス
RemotePort\$	Trap を発生したマシンのポート番号
Version%	SNMP のバージョン番号(0:Version1 固定)
Community\$	コミュニティ(パスワード)
Enterprise\$	この Trap を定義した登録組織のオブジェクト ID
AgentAddress\$	トラップの送信元の IP アドレス
GenericTrapType%	イベントの種類
SpecificType\$	特殊なトラップの場合の詳細な情報(整数)
TimeStamp\$	経過時間(1/100 秒)
ObjectCollection _ As Collection	オブジェクト名とその値と型の情報の集合
RawData\$	生データ(バイナリ文字列)

ObjectCollection は、SnmTrapObject の集合です。

ObjectCollection の内容を確認するには以下のようにコーディングします。

```
Dim v As SmnpTrapObject
For Each v In ObjectCollection
    Debug.Print "Object ID=" & v.OIDName
    Debug.Print "Object TYPENAME=" & v.OIDTypeName
    Debug.Print "Object TYPE=" & v.OIDType
    Debug.Print "Object VALUE=" & v.OIDValue
Next
```

あるいは下記のようにコーディングします。

```
Dim i&
For i = 0 To ObjectCollection.Count - 1
    Debug.Print "Object ID=" & ObjectCollection (CStr(i)).OIDName
    Debug.Print "Object TYPENAME=" & ObjectCollection (CStr(i)).OIDTypeName
    Debug.Print "Object TYPE=" & ObjectCollection (CStr(i)).OIDType
    Debug.Print "Object VALUE=" & ObjectCollection (CStr(i)).OIDValue
Next
```

白紙ページ

## ▼ プロパティ

白紙ページ

# Stat プロパティ

---

## ■機 能

SNMPTRAP コントロールの状態を示します。

## ■構 文

Object.Stat

Stat 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	SNMPTRAP コントロールオブジェクトです。

## ■データ型

整数(Integer)。

## ■戻り値

(定 数)	(説 明)
-1	初期状態です。(受信不可)
0	受信開始待ちです。(受信不可)
1	受信待ちです。(受信可能)

## ■解 説

StartTrap を行うと、Stat は 1 になります。

StopTrap を行うと、Stat は-1 になります。

Stat が 0 から変化しない場合、Winsock に問題がある可能性があります。

# StringType プロパティ

---

## ■機能

受信した OctedString の取り扱いを指示します。

## ■構文

Object.StringType [=Value]

StringType 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	SNMPTRAP コントロールオブジェクトです。
Value	文字列を Unicode に変換するかどうかを指示する値です。

## ■データ型

長整数(Long)。

## ■設定値

Value の設定値は次の通りです。

(定数)	(説明)
0	初期状態。Unicode に変換します。
1	Unicode に変換しません。

## ▼ メソッド

白紙ページ



# StartTrap メソッド

---

## ■機能

Trap 受信を開始します。

## ■構文

Object.StartTrap([Port As String = "162"])

StartTrap 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	SNMPTRAP コントロールオブジェクトです。
Port	受信ポートです。通常は 162 です。

## ■戻り値

なし

## ■解説

StartTrap を行うと TRAP PDU の受信が可能になります。

すでに StartTrap 済みである場合、何もしません。

同じ PC で受信しているプログラムがあると、トラップ可能なエラーが発生します。

(Err.Number=10048)

# StopTrap メソッド

---

## ■機 能

Trap 受信を終了します。

## ■構 文

Object.StopTrap()

StopTrap 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	SNMPTRAP コントロールオブジェクトです。

## ■戻り値

なし

## ■解 説

StopTrap を行うと TRAP PDU の受信を終了します。  
受信を開始していない場合、何もしません。

## ▼ イベント

白紙ページ

# Ready イベント

---

## ■機 能

Trap 受信が可能になったときに発生します。

## ■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	SNMPTRAP コントロールオブジェクトです。

## ■解 説

StartTrap メソッドを実行した結果、受信可能状態になったことをプログラムに通知します。

# Trap イベント

## ■機能

TRAP PDU を受信したときに発生します。

## ■構文

Private Sub Object\_Trap

(ByVal RemoteIP As String, ByVal RemotePort As String, ByVal Version As Integer, ByVal Community As String, ByVal Enterprise As String, ByVal AgentAddress As String, ByVal GenericTrapType As Integer, ByVal SpecificType As String, ByVal TimeStamp As String, ObjectCollection As Collection, ByVal RawData As String)

Trap 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	SNMPTRAP コントロールオブジェクトです。
RemoteIP	TRAP PDU を送信した IP です。
RemotePort	TRAP PDU を送信したポート番号です。
Version	SNMP のバージョンです。(0:Version1 固定)
Community	コミュニティ名です。
Enterprise	TRAP を定義した企業のオブジェクト ID です。
AgentAddress	TRAP の原因となった機器の IP アドレスです。 (通常 RemoteIP と一致)
GenericTrapType	イベントの種類です。
SpecificType	特殊なトラップの場合の詳細な情報です。(整数)
TimeStamp	トラップ発生時間です。(1/100 秒)
ObjectCollection	オブジェクト名とその値と型の情報の集合です。
RawData	生データです。(バイナリ文字列)

## ■解説

TRAP PDU を受信したとき、Trap イベントが発生します。

ObjectCollection は、SnmptTrapObject の集合です。

SnmptTrapObject には、4つのメンバーがあります。

OIDName	オブジェクト ID
OIDTypeName	オブジェクトの型を示す文字列
OIDType	オブジェクトの型を示す整数
OIDValue	オブジェクトの値 (Variant)

オブジェクトの型を示す文字列と整数の関係は、以下のとおりです。

(整数)	(名前)
------	------

2	Integer
4	OctetString
5	Null
6	ObjectID
&H40	IpAddress
&H41	Counter
&H42	Gauge
&H43	TimeTicks
&H44	Opaque
Else	数値そのまま

ObjectCollection の内容を確認するには以下のようにコーディングします。

```
Dim v As SnmpTrapObject
For Each v In ObjectCollection
    Debug.Print "Object ID=" & v.OIDName
    Debug.Print "Object TYPENAME=" & v.OIDTypeName
    Debug.Print "Object TYPE=" & v.OIDType
    Debug.Print "Object VALUE=" & v.OIDValue
Next
```

あるいは下記のようにコーディングします。

```
Dim i&
For i = 0 To ObjectCollection.Count - 1
    Debug.Print "Object ID=" & ObjectCollection (CStr(i)).OIDName
    Debug.Print "Object TYPENAME=" & ObjectCollection (CStr(i)).OIDTypeName
    Debug.Print "Object TYPE=" & ObjectCollection (CStr(i)).OIDType
    Debug.Print "Object VALUE=" & ObjectCollection (CStr(i)).OIDValue
Next
```

# WsError イベント

---

## ■機 能

SNMPTRAP コントロール内で Winsock エラーを検出した場合に発生します。

## ■構 文

Private Sub Object\_WsError (Ecode As Long, Description As String, Where As String)  
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	SNMPTRAP コントロールオブジェクトです。
Ecode	Winsock エラーコードです。
Description	エラーの説明です。
Where	エラー発生場所です。

## ■解 説

問題解決のため、これらの情報を記録してください。



# サンプル

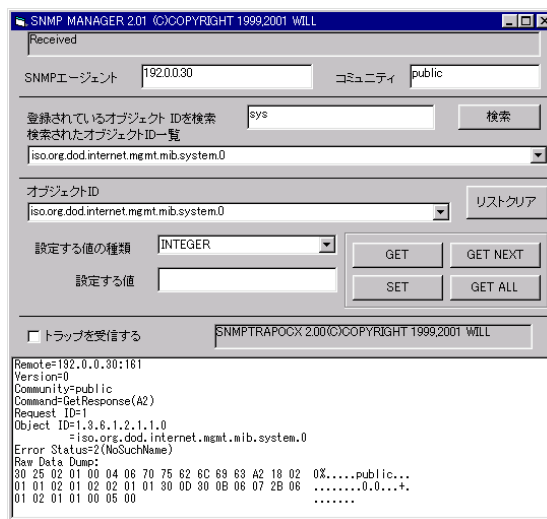
サンプル

白紙ページ

# SNMP Manager サンプル (SNMP MANAGER 2.01)

SNMP MANAGER 2.01 は、SNMPOCX を用いたサンプルプログラムです。以下の2つの機能があります。

- MIB 表示・設定機能  
SNMP エージェントに存在する MIB の値を簡単に検索、設定。
- トラップの受信機能  
トラップの受信。



## ■ MIB 表示・設定機能

SNMP エージェントに存在する MIB の値を簡単に検索し設定できます。以下の手順に従ってサンプルをご使用ください。

### 1. SNMP エージェントに接続

- SNMP エージェント  
SNMP エージェントの IP アドレス等を入力します。
- コミュニティ  
SNMP エージェントに接続する時のパスワードを入力します。(通常は「public」です。)

### 2. オブジェクトの検索

- 登録されているオブジェクト ID  
検索したいオブジェクトを入力し「検索」ボタンを押します。「一覧」にオブジェクトの一覧が表示されます。(例：SYS 等)
- 検索されたオブジェクト ID 一覧  
検索されたオブジェクトの ID 一覧が表示されます。(検索項目に何も入力しない場合は、デフォルトで設定されている一般的なオブジェクト ID が表示されます。)

### 3. オブジェクト内容の表示・設定

- オブジェクト ID  
一覧表の中から、変更したいオブジェクトを選択し「GET」ボタンを押します。この時「GET ALL」ボタンを押すと SNMP エージェント内の MIB オブジェクトがすべて表示されます。

## サンプル

選択されたオブジェクトが存在しない場合は、「Error Status=2(NoSuchName)」というメッセージが表示されます。

オブジェクト ID の最後の数字が「0」の場合は、そのオブジェクトの中に、それ以上オブジェクトが含まれていないことを示します。また、「1」以上の場合は、さらにオブジェクトが含まれていることを示しています。この場合、「GET NEXT」ボタンで検索します。

選択したオブジェクトは、すべて「オブジェクト ID」リストの中に記録されます。記録を削除したい場合は、「リストクリア」ボタンを押します。

- **設定する値の種類**

オブジェクトの内容の「Object Type」を参照し選択します。

- **設定する値**

「Value」の内容を変更します。設定したい値を入力し「SET」ボタンを押します。

## ■ トラップ受信機能

「トラップを受信する。」のチェックボックスにチェックをつけると、SNMP エージェントからの TRAP PDU を受信できます。

# SENDTRAP サンプル

## SendTrap メソッド 2.00

SNMP マネージャーに TRAP を通知するサンプルです。

以下の手順に従って、ご利用ください。

※「SNMP MANAGER 2.01」と併せて利用できます。

[TRAP種別]	
一般トラップタイプ	EnterpriseSpecific
特別トラップタイプ	1
企業ID	1.3.6.1.4.1.5000

[通知データ]	
オブジェクトタイプ	文字列(OctetString)
オブジェクトID	1.3.6.1.4.1.5000.1
オブジェクト値	123456

[TRAP受信サーバー]	
ホスト名(IPアドレス)	127.0.0.1
コミュニティ	public

送信

### 1. TRAP 種別

- 一般トラップタイプ  
リストボックスにある7種類のトラップの中からからいずれかを選択します。
- 特別トラップタイプ  
任意の数値を記入します。
- 企業 ID  
TRAP を発信する企業のオブジェクト ID を指定します。

### 2. 通知データ

- オブジェクトタイプ  
データを通知する時のオブジェクトタイプを指定します。
- オブジェクト ID  
トラップを発信するオブジェクト ID を指定します。
- オブジェクト値  
通知する値を指定します。

## サンプル

### 3. TRAP 受信サーバー

- ホスト名 (IP アドレス)  
受信する SNMP マネージャーの IP アドレスを指定します。
- コミュニティ  
受信する SNMP マネージャーのパスワードを指定します。

上記の各項目の設定が完了したら、「送信」ボタンをクリックして SNMP マネージャーに通知データが送信されることを確認して下さい。

## 索引

AgentAddress プロパティ .....	33
ClearOID メソッド .....	40
Err2Str メソッド .....	39
FindName メソッド .....	44
Name2OID メソッド .....	43
Name2TypeID メソッド .....	45
OID2Name メソッド .....	42
Ready イベント .....	63, 81
Response イベント .....	61
SendGetNextRequest メソッド .....	55
SendGetRequest メソッド .....	54
SendSetRequest メソッド .....	56
SendTrap メソッド .....	57
SetOID メソッド .....	41
SNMPBinStringValue メソッド .....	53
SNMPCounterValue メソッド .....	47
SNMPGaugeValue メソッド .....	51
SNMPIntegerValue メソッド .....	48
SNMPIpAddressValue メソッド .....	50
SNMPNullValue メソッド .....	46
SNMPStringValue メソッド .....	49
SNMPTimeTicksValue メソッド .....	52
StartTrap メソッド .....	77
Stat プロパティ .....	31, 73
StopTrap メソッド .....	78
StringType プロパティ .....	35, 74
TickCount プロパティ .....	34
Trap イベント .....	82
WsError イベント .....	64, 84

白紙ページ



## SNMPOCX マニュアル

1998年	1月31日	初版	第1版
1999年	6月1日		第2版
1999年	7月1日		第3版
2001年	6月15日		第4版
2001年	12月10日		第5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。

---