

第2章

2

SNMPマネージャ開発

見本

■ 概要

SNMPは、TCP/IPベースのネットワークで、コンピュータ・ハブ・ブリッジ・ルータなどのネットワーク機器状態とネットワークトラフィックを収集するためのプロトコルです。SNMPは、ネットワーク上でデータを蓄積する「SNMPエージェント」と、それらのデータを収集・管理する「SNMPマネージャ」で構成されます。

本章では、Windows NT/2000/XPでSNMPマネージャを開発する方法について説明します。SNMPマネージャは、ネットワーク管理者・システム管理者の監視用PC上で動作するアプリケーションプログラムです。市販ソフトウェア製品ではHP Openview、Novell NMS、IBM NetView、Sun Net Managerなどが有名です。SNMPマネージャとSNMPエージェントとの間では、情報取得・情報設定・事象通知という3つの動作があります。

■ 情報取得

監視用PC上で動作するSNMPマネージャから、ネットワーク機器上で動作するSNMPエージェントに対して情報取得(Get/GetNext)が要求されると、SNMPエージェントは管理しているMIBの情報を、SNMPマネージャに返却(GetResponse)します。

■ 情報設定

SNMPマネージャから、SNMPエージェントに対して情報設定(Set)が要求されると、SNMPは管理しているMIBの情報を更新し、SNMPマネージャに結果(GetResponse)を報告します。

■ 事象通知

SNMPエージェントで、なんらかの事象が発生した際には、SNMPマネージャに対して、情報(Trap)を送信します。事象通知は、情報取得や情報設定とは異なり、SNMPエージェントから自律的に送信されます。

2.1

情報収集(Get)の開発例

2

2.1.1 開発例の概要

概要

ここでは、SNMPマネージャからSNMPエージェントに対して、Getメッセージを送信して情報収集するプログラムを開発する例について説明します。開発・実行には、1台のマシンを使用し、OSにはWindows 2000 Professionalを、開発ソフトウェアにはVisual C++ 6.0(C言語)とPlatform SDKのSNMP Functionsを使用します。また、Windows 2000 ProfessionalにはSNMP Serviceがインストール・設定されて、開始している必要があります。必要なソフトウェアがインストールされていることを確認してください。

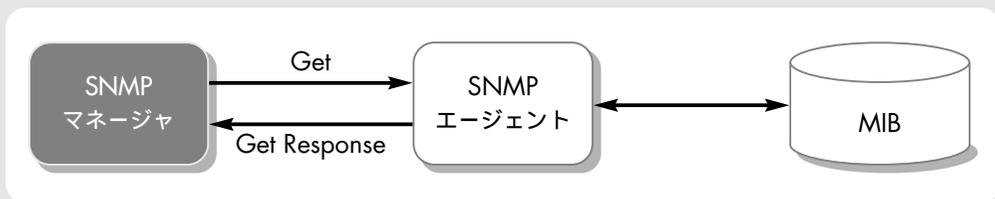
注意事項

SNMP Serviceのインストールと設定については、本書「1.3 WindowsでのSNMP実装」を参照してください。Platform SDKの概要、インストール方法については、本書「付録1 Platform SDK」を参照してください。

開発・実行に2台のマシンを使用し、SNMPマネージャとSNMPエージェントを別々のマシン上で実行してもかまいません。また、SNMPエージェントにWindowsマシンでなくネットワーク機器(ルータなど)を使用してもかまいません。

この開発例では、SNMPマネージャがどのようにSNMPエージェントにリクエストを発行するか、Getメッセージをどのように送信するか、GetResponseメッセージの内容をどのように参照するかなどがわかります。開発例の概要を、図2-1に示します。

図2-1 情報収集(Get)の開発例概要



- Windows 2000 Professional (SNMP Service)
- Visual C++ 6.0
- Platform SDK

開発の手順

開発のおおまかな流れは、以下のとおりです。

2

プロジェクトの作成
プロジェクトの設定
ソースコードの入力
コンパイル・実行

2.1.2 プロジェクトの作成

まずはじめに、VC++を起動してプロジェクトの新規作成をおこないます。以降の説明では、プロジェクト名は、「MgrGet」となっています。

1. [スタート]メニュー [プログラム] [Microsoft Visual Studio 6.0] [Microsoft Visual C++ 6.0] を順にクリックし、VC++を起動します。
2. VC++の[ファイル]メニューの[新規作成]をクリックします。
3. 新規作成ダイアログで、「Win32 Console Application」を選択し、位置とプロジェクト名(MgrGet)を入力し、[OK]ボタンをクリックします。(図 2-2)
4. 「Win32 Console Application ステップ1/1」で、「空のプロジェクト」がチェックされているのを確認し、[終了]ボタンをクリックします。(図 2-3)
5. 新規プロジェクト情報が表示されたら、[OK]ボタンをクリックします。(図 2-4) プロジェクトが新規に作成されます。

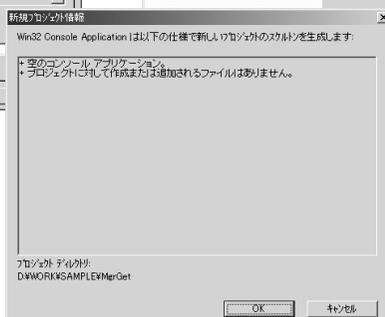
図 2-2 新規作成ダイアログ



図 2-3 「Win32 Console Application ステップ1/1」



図 2-4 新規プロジェクト情報



2.1.3 プロジェクトの設定

プロジェクトの作成が完了したら、ライブラリファイル・インクルードファイルなど、プロジェクトの設定をおこないます。

2

1. VC++のワークスペースウィンドウ(通常は左側に表示)で,[FileView]タブをクリックし、「MgrGetファイル」をダブルクリックします。(図2-5)
2. 「MgrGetファイル」を右クリックし,[フォルダの新規作成]をクリックします。
3. フォルダの新規作成ダイアログで,新しいフォルダの名前に「Library Files」、ファイルの拡張子に「lib」と入力し,[OK]ボタンをクリックします。(図2-6)「MgrGetファイル」の配下に「Library Files」フォルダが作成されます。
4. 「Library Files」フォルダを右クリックし,[ファイルをフォルダへ追加]をクリックします。
5. プロジェクトへファイルを追加ダイアログで,ファイルの種類を「ライブラリファイル(.lib)」に変更し,ファイルの場所を「Platform SDKのライブラリがインストールされているフォルダ(通常は、¥Program Files¥Microsoft SDK¥Lib)」に移動し,MgmtAPI.libを選択し,[OK]ボタンをクリックします。(図2-7) MgmtAPI.libは,SNMPマネージャを開発する際に使用するライブラリです。
6. 再び,「Library Files」フォルダを右クリックし,[ファイルをフォルダへ追加]をクリックします。
7. プロジェクトへファイルを追加ダイアログで,ファイルの種類を「ライブラリファイル(.lib)」に変更し,ファイルの場所を「Platform SDKのライブラリがインストールされているフォルダ(通常は、¥Program Files¥Microsoft SDK¥Lib)」に移動し,SnmpAPI.libを選択し,[OK]ボタンをクリックします。(図2-8) SnmpAPI.libは,SNMPマネージャ・エージェントどちらの開発

図2-5 ワークスペースウィンドウのFileViewタブ



図2-6 フォルダの新規作成

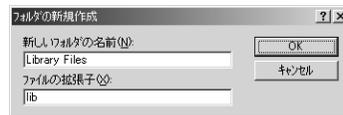


図2-7 プロジェクトへファイルを追加

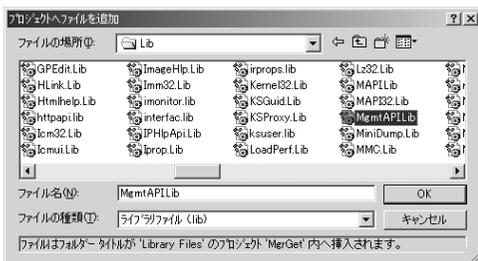
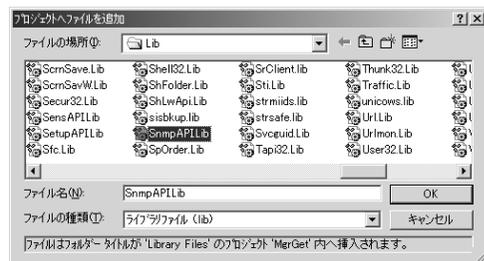


図2-8 プロジェクトへファイルを追加



でも使用するライブラリです。

8. ここまでの設定で、VC++のワークスペースウィンドウは、図2-9のようになります。
9. 次に、プロジェクトにソースファイルを追加します。VC++の [プロジェクト] メニュー [プロジェクトへ追加] [新規作成] をクリックします。(ソースファイルは、VC++以外のテキストエディタで作成し、VC++のプロジェクトに追加してもかまいません。)
10. 新規作成ダイアログが表示されたら、「C++ ソースファイル」を選択し、ファイル名に「MgrGet.c」と入力し、[OK] ボタンをクリックします。(図2-10)
11. ソースファイル(MgrGet.c)が表示されたら、ソースコードを入力します。ソースコードは、図2-11～17です。

2.1.4 ソースコードの解説

概要

この開発例のソースコードは、以下のパートに分かれています。

インクルードファイル・ワークエリア
 SNMP関連の初期化処理
 変数バインディングの編集
 SNMPマネージャAPIのオープン
 指定操作(Get)を要求
 指定操作の応答(GetResponse)を表示
 SNMP関連の終了処理

この開発例では、SnmpMgrOpen関数 SnmpMgrRequest関数 SnmpMgrClose関数を順番に呼び出してGetメッセージによる情報収集を実現しています。その他のAPIは、SNMP固有のデータを変換するためのユーティリティAPIです。この開発例で使用するSNMP FunctionsのAPIを、表2-1に示します。

図2-9 ワークスペースウィンドウ



図2-10 新規作成ダイアログ

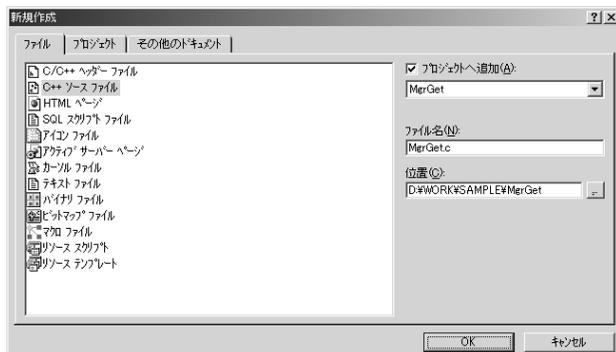


表2-1 使用するSNMP Functions

分類	API名	説明
SNMP マネージャ API	SnmpMgrOpen	通信ソケットと関連データ構造を初期化する
	SnmpMgrRequest	指定された操作(GetRequest, GetNextRequest, Set Request)の実行を要求し, 応答(GetResponse)を受信する
	SnmpMgrClose	通信ソケットと指定されたSNMPセッションに関連したデータ構造をクローズする
	SnmpMgrOidToStr	OIDを文字列に変換する
	SnmpMgrStrToOid	文字列をOIDに変換する
SNMP ユーティリティ API	SnmpUtilMemAlloc	メモリを割り当てる
	SnmpUtilMemReAlloc	メモリを再割り当てする
	SnmpUtilMemFree	メモリを解放する
	SnmpUtilOidFree	OIDのデータエリアを解放する
	SnmpUtilPrintAsnAny	AsnAny構造体型のデータを標準出力に表示する
	SnmpUtilVarBindListFree	変数バインディングリストのデータエリアを解放する

2

インクルードファイル・ワークエリア

ソースコードの1行目から32行目では、インクルードファイルの宣言、ワークエリアの定義をおこなっています。

図2-11 ソースコード(1行目~32行目)

```
// =====
// <MgrGet.c> SNMP マネージャ (Get) サンプルアプリケーション
// =====
// -----
//   インクルードファイル
// -----
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include <snmp.h>
#include <mgmtapi.h>

// -----
//   ワークエリア
// -----
INT     timeout  = 6000;           // タイムアウト値(ミリ秒)
INT     retries  = 3;             // リトライ回数
```