

SNMPC 7.1

リモートアクセス拡張
カスタムサービスポーリング
カスタムスクリプト

Castle Rock Computing
September, 2007

デジタルアーキテクト株式会社

概要

このドキュメントでは SNMPc バージョン 7.1 の新しい機能について説明します。

リモートアクセス拡張

SNMPc 7.1 エンタープライズ版では 10 リモートポーラ、10 リモートコンソール、および無制限 JAVA コンソールを基本システムに含みます。エンタープライズ版のオプションであるリモートアクセス拡張版ライセンスは、無制限のリモートポーラおよび無制限のリモートコンソールを許可します。

カスタムサービスポーリング

SNMPc 7.1 はユーザの開発したアプリケーションまたはスクリプトを使用できるサービスポーリング拡張機能を提供します。カスタムサービスポーリングを使用して SNMP 非対応や IP プロトコル非対応の古いレガシーシステムにポーリングを行うことができます。また、ネットワークデバイスの故障や劣化について詳細にわたる分析を行い、管理者に問題の根本原因および推奨される対策方法などの情報を提供します。

カスタムスクリプトのサポート

SNMPc 7.1 ではどのスクリプト言語からも使用できる新しい機能が含まれ、スクリプトベースのアプリケーションを開発することができます。プログラミング言語ではなくスクリプトを使用することでカスタムアプリケーションの開発や既存の管理環境との統合を簡素化します。また、トラップ受信時、マップアイコンをダブルクリックしたとき、またはカスタムメニューよりおよびカスタムポーリングアプリケーションとしてカスタムスクリプトを実行することができます。

全てのカスタムスクリプトサポートコマンドのソースファイルは *SDK\scripts* サブディレクトリに含まれます。

カスタムサービスポーリング

カスタムサービスポーリングは SNMPc *TCP Service Polling* モジュールに追加されました。今後は主に異なるサービスプロトコルをサポートするためモジュールは *Service Polling* にリネームされました。カスタムポーリングアプリケーションは管理サーバまたはリモートコンピュータシステム内にある SNMPc ポーリングエージェントによって実行されます。

カスタムサービスはマップアイコンプロパティの *Service Polling* 属性で指定されます。それぞれに *Service Name* および *Polling Application* と利用できるアプリケーション引数を持ちます。ポーリングアプリケーションは通常の Windows で実行または Perl, Tcl, PHP などを使用したスクリプトで実行することができます。

スクリプトベースのポーリングアプリケーションには、以下のような共通の操作を実行するための新しいサポート機能が利用可能になりました。

- SNMP Get, Set および Trap
- SNMPc マップアイコン属性の取り込みおよび変更
- SNMPc マップの Get/Set アプリケーション詳細データ

ポーリングアプリケーションはポーリング結果を 0 から 100 までの数字でかえします。数字は *Quality of Service* 測定基準をワーストからベストで表し、0 は完全なポーリング失敗を示します。この値は SNMPc では二箇所で使用されます。まず、ゼロまたは非ゼロの値は *responding/notResponding* サービスポーリングステータスを示し、通常の *Service State* イベントおよび *Availability* レポートのどちらを使用しているかをレポートします。次に、*Availability* テーブル内の *Quality of Service* (QOS) と名づけられた新しいカラムで値が表示されます。QOS 値は SNMPc テーブル、グラフ、および長期トレンドレポートに使用することができます。

カスタムポーリングアプリケーションはポーリング状態の詳細情報を提供するため SNMP トラップを生成することが一般に予想されます。これらのトラップは SNMPc *System Info* トラップ定義またはユーザ定義のベンダ独自トラップを使用することができます。関連のデバイスにポーリングを行い、ネットワーク状態の詳細な分析を行うことでカスタムポーリングアプリケーションはネットワーク問題におけるハイレベルな解決策を提供します。例えば、デバイス故障問題だけでなくデバイスパフォーマンスの深刻な低下問題があげられます。

新 SNMP MIB テーブルと変数

以下の新しい SNMP mib テーブルが追加されました。

```
private/CastleRock/snmpc/availGroupV7/AvailEntryQos  
private/CastleRock/snmpc/availGroupV7/AvailReportEntryQos
```

どちらのテーブルも *AvailReportEntry* テーブルに似た形式ですが新しい変数 *availQualityOfService* が追加されています。この新しい変数は *AvailEntryQOS* テーブルの *availAveragePercentUptime* を置き換えます。新しい *availQualityofService* 変数の定義を以下に記します。

- **SNMPc でポーリングされたサービス**, (通常の ICMP および SNMP ポーリングを含む) Quality of Service は Percent Uptime の移動平均で最終の 20 ポーリングサンプルを使用します。
- **カスタムアプリケーションでポーリングされたサービス**, Quality of Service は計算された値でアプリケーションから戻ってくる値です。0 は失敗を意味し 100 はベストサービス品質を意味します。SNMPc はこの値を *Responding* (1...100) および *notResponding* (0) サービスステータスにマップします。

ポーリングアプリケーションユーザインタフェース

ポーリングアプリケーションの設定する場合にはマップアイコンを右クリックして *Edit Properties* メニューを使用します。次に *Attributes* タブを選択して *Service Polling* 属性をダブルクリックします。

Poll Exec 新規ボックスが *Edit Services* ダイアログの一番下に追加されます。ポーリングアプリケーションを実行するコマンドラインまたはスクリプトおよび必要な引数を入力します。次のパラメータをコマンドラインに含むことができます。

- ***\$i*** ポーリングオブジェクトの SNMPc マップレコード番号。この識別子は関連するマップ情報にアクセスするのに使用して SNMP オペレーションを実行できます。
- ***\$a*** ポーリングオブジェクトのインターネットアドレス。TCP,UDP またはその他独自管理プロトコルに使用することができます。
- ***\$c*** ポーリングオブジェクトの SNMPc *Access Parameter* 属性。この文字列は SNMP セキュリティクレデンシャルをエンコードし、Get、Set またはトラップコミュニティ名として SNMPc プログラミングインタフェースと使用することができます。

通例、各ポーリングアプリケーションには *\$i* または *\$a* のどちらかのパラメータが必要です。

スクリプトを使用する場合は、スクリプト解釈をコマンドとして指定し、スクリプト名を引数のひとつとして含む必要があるかも知れません。Visual Basic スクリプトのコマンドラインの例:

```
"cscript.exe pollapp.vbs $!"
```

NOTE: コマンドラインにフルパス名を含む場合は、プログラムはどのリモートポーリングエージェントシステムでも全く同じドライバ/ディレクトリパスにおかれている必要があります。フルパス名を含まない場合、コマンドは SNMPc インストールディレクトリ、SNMPc サブディレクトリ、または Windows *PATH* 環境変数ディレクトリの中のひとつにおくことができます。

ダイアログの右側にある **Add** ボタンで *All Services* リストに新しいサービス定義を追加します。新しいサービス名を追加した後は *All Services* リストボックスから異なるマップアイコンに再利用することができます。

Test ボタンで SNMPc コンソールからアプリケーションを実行します。結果のトレース情報はダイアログボックスに表示されます。

ポーリングアプリケーション機能が正しく動作していることを確認後、ダイアログ上部の **Add>>** ボタンでこのサービスを選択マップオブジェクトのポーリングサービスリストに追加します。

ポーリングアプリケーションの実装

ポーリングアプリケーションは C++などのプログラミング言語や DOS バッチファイル、Perl、Tcl などのスクリプト言語で開発することができます。ポーリングアプリケーションで利用できるインタフェース引数はデバイスマップレコード番号、IP アドレスおよびマップアクセスパラメータです。

各ポーリングアプリケーションはデバイスで実行されている特定のサービスについて *Quality of Service* を決定します。この測定基準は失敗を意味する 0 からベストサービス品質を意味する 100 までの数字で表します。数字はプログラムまたはスクリプトの戻り値として SNMPc に送信されます。測定が 88 の場合の例を以下に記します。

- *DOS Batch File (.bat):* exit 88
- *Visual Basic Script (.vbs):* WScript.Quit 88
- *Perl Script (.pl):* exit (88)
- *C/C++ Program (.exe):* return (88);

ポーリングアプリケーションで使用される通信メカニズムおよびアルゴリズムについては開発者が全て定義し、SNMPc の範囲外となります。

次のカスタムスクリプトのサポートセクションでは、スクリプトベースのポーリングアプリケーション開発に利用できる新しいコマンドを説明します。“C”または“C++”を使用する SNMPc アプリケーションの開発についての詳細は SNMPc *SDK\DOCS* サブディレクトリの *SNMPc C/C++ Programming Interface* ドキュメントを参照してください。

カスタムスクリプト

SNMPc では、サードパーティによる SNMPc の拡張機能およびシステムインテグレーションモジュール開発のための *Applications Programming Interface* をサポートしてきました。しかしながら、従来の C++ や Visual Basic 等のプログラミング言語を使用してのアプリケーション開発は長期間におよび、コストのかかる作業でした。

カスタムアプリケーション開発を簡素化するため、SNMPc7.1 では DOS バッチファイル、Perl、Tcl やその他スクリプト言語を使用できる機能を提供します。

カスタムスクリプトは Windows プログラムを使用できるモジュールであれば SNMPc から実行することができます。例えば、

- ・ トラップを受信したとき
- ・ マップのデバイスアイコンをダブルクリックしたとき
- ・ SNMPc カスタムメニューより
- ・ カスタムポーリングアプリケーションとして

カスタムスクリプトアプリケーションに使用可能な引数はスクリプトを実行させる方法に依存することに注意してください。様々なアプリケーション実行のモジュールにて使用可能な引数については該当する SNMPc ドキュメントを参照ください。

SNMPc7.1 ではカスタムスクリプトに使用するため新しいコマンドが幾つか追加されました。これらのコマンドはデバイス SNMP オペレーション、クエリ、マップデータおよびトレンドレポートの設定等様々な共通機能を実行することができます。全てのカスタムスクリプトコマンドのソースファイルは SNMPc の *SDK\scripts* サブディレクトリに含まれます。

全てのスクリプトコマンドは次の *Server Access Parameters* オプションを使用できます。

- ・ *-a Address* SNMPc サーバの IP アドレス
- ・ *-u User* サーバにログインする際に使用するユーザ名
- ・ *-p Password* サーバにログインする際に使用するパスワード

Address パラメータが指定されていない場合、ローカルで実行されているポーリングエージェントに使用されているサーバに接続されます。*User* および *Password* が指定されていない場合、*Remote Poller* のクレデンシャル設定が使用されます。

出力結果は *Name:Attribute=Value* の形式で表示されます。これは *-b* パラメータオプションを使用して *Value* だけに省略することができます。エラーやその他の失敗にはテキストの始まりに *Error:* を含むエラーメッセージが出力されます。

使用可能なスクリプトコマンドおよび引数は次のセクションで説明します。
スペースを含むコマンド引数は、引数を“二重引用符”で囲みます。

デバイス SNMP コマンド

SNMPGET [Acc] [-n] [-b] Recno¥Name Variable1[Variable2...]

上記コマンドは、単数または複数変数の SNMP *Get* または *GetNext* 作業を実行します。出力結果は作業結果で *NodeName:Variable=Value* の形式で変数と値は別のラインに出力されます。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-n* SNMP *GetNext* オペレーションを使用する
- ・ *-b* Bare モード; 値のみ表示
- ・ *Recno |Name* マップオブジェクトの番号または名前を記録する
- ・ *VariableN* SNMP 変数名

SNMPSET [Acc] [-b] Recno ¥Name Variable1 Value1 [Variable2 Value2...]

上記コマンドを使用して SNMP *Set* 操作を単数または複数変数に行います。出力は各変数および値を別々の行にて *NodeName:Variable=Value* の形式で操作結果をかえします。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-b* Bare モード; 値のみ表示
- ・ *Recno |Name* マップオブジェクトの番号または名前を記録する
- ・ *VariableN* 設定する SNMP 変数
- ・ *ValueN* 設定する SNMP 値

SNMPTRAP [Acc] Recno ¥Name enterprise TrapOID [Var1 Val1 [Var2 Val2...]]

上記コマンドを使用して SNMP トラップ操作を行います。このコマンドは *Recno |Name* で指定するデバイスの *Proxy* として機能します。すなわち、SNMPc で表示されるトラップは指定されたマップオブジェクトから発生します。このコマンドは成功したときの出力はありません。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *Recno |Name* マップオブジェクトの番号または名前を記録する
- ・ *Enterprise* SNMP トラップ エンタープライズオブジェクト識別子
- ・ *TrapOID* SNMP トラップオブジェクト識別子
- ・ *VarN* SNMP トラップ引数
- ・ *ValN* SNMP トラップ引数値

マップデータベースコマンド

MAPFIND [Acc] [-b] Key Value

キーの値をもとにマップオブジェクトを探します。多数のオブジェクトが同じキーの値に一致する場合は、最初に一致する値がかえってきます。マップレコード数が出力され、*NodeName:RECNO=recno* の形式で表示されます。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-b* Bare モード; レコード数のみ表示
- ・ *Key* キーを検索; *Name*、*Parent*、*Type*、*Address* のうちひとつ
- ・ *Value* 完全に一致する値を検索します。 *Parent* はマップレコード数; ルートサブマップの *Parent* 値として 0 を使用します。 *Type* は、デバイス、サブネット、*Goto*、リンク、ネットワーク、リングネットワーク、バスネットワークのうちひとつ

MAPNEXT [Acc] [-b] Recno Key

Recno で指定されるオブジェクトと同様の名付けられたキーの値に一致する次のマップオブジェクトを見つけます。このコマンドは頻繁に全ての同じタイプのオブジェクトのスキャンまたは同じサブネット内の全てのオブジェクト(例: 同じ Parent)のスキャンに使用されます。

NodeName:RECNO=recno の形式でマップレコード番号が出力されます。

- *Acc* SNMPc Server アクセスパラメータ
- *-b* Bare モード;レコード番号のみ表示
- *Recno* マップオブジェクトのレコード番号; *MAPFIND* よりかえってくる
- *Key* 検索キー; *Name*、*Parent*、*Type*、*Address* のうちひとつ

MAPADD [Acc] [-b] Type Parent Name IpAddr [X Y | Link1 Link2]

指定するオブジェクトタイプをマップに追加します。*NodeName:RECNO=recno* の形式で新しいオブジェクトのマップレコード番号が出力されます。

- *Acc* SNMPc Server アクセスパラメータ
- *-b* Bare モード;レコード番号のみ表示
- *Type* オブジェクトタイプ; デバイス、サブネット、*Goto*、リンク、ネットワーク、リングネットワーク、バスネットワークのうちひとつ
- *Name* マップオブジェクト名
- *Parent* 新しいオブジェクトが追加されるべきサブマップ下の名前またはレコード番号。ルートサブマップには 0 を使用します。
- *IpAddr* マップオブジェクトの IP アドレス(または 0.0.0.0)
- *X, Y* 任意の X と Y 位置。マップの中心は 0, 0 です。
- *Link1 Link2* 任意のレコード番号またはリンクが接続されている二つのオブジェクト名。X,Y 位置変数の代わりに使用します。

MAPATTR [Acc] [-b] Recno |Name ATTR |UAATTR ` [Value]

ポーリングオブジェクトのマップアイコンからまたはマップアイコンへ情報を読み取り/書き込みます。マップの属性値または書き込みの場合、プログラムの出力は新しい値となります。属性を特定せずに実行した場合、*NodeName:AttrName=Value* の形式で全てのノード属性名および値を表示します。

- *Acc* SNMPc Server アクセスパラメータ
- *-b* Bare モード;レコード番号のみ表示
- *Recno |Name* マップオブジェクトのレコード番号または名前
- *ATTR* 読み取りまたは書き込み属性
- *'UAATTR'* 読み取りまたは書き込みのカスタム属性。カスタム属性は 31 バイトまでの長さのテキストを引用符(“)で囲みます。カスタム属性の意味およびそれに対応する値はスクリプト開発者が定義します。
- *Value* 属性を書き込みする値を指定

次のテーブルは *MAPATTR* 機能で使用できるビルトイン属性を簡単に説明します。これらの属性についての詳細は、*SDK/DOCS* サブディレクトリの *SNMPc Application Programming Interface* ドキュメントを参照してください。

属性の名前	説明
RECNO	ノードのレコード番号を取得します。ノード名よりレコード番号を検索します。(読み取りのみ)
NAME	オブジェクト名
DESCR	説明
TYPE	オブジェクトタイプ: デバイス、サブネット、Goto、リンク、ネットワーク、リングネットワーク、バスネットワーク
PARENT	オブジェクトのあるサブネットのレコード番号
ICON	アイコンファイル
STATUS	カレントオブジェクトステータス(読み取りのみ)
EXEC	実行するプログラムをダブルクリック
ADDRESS	IP アドレス
MACADDR	MAC アドレス
BITMAP	サブネットの背景ビットマップ
BKSHAPE	オブジェクトの背景の形: 四角形、円、ダイヤモンド、六角形、八角形
BITMAPSCALE	背景ビットマップスケール
SHOWNAME	オブジェクトのラベルをマップに表示
POLLINT	ポーリングインターバル (秒)
POLLRETRY	ポーリング再試行
POLLTIMO	ポーリングタイムアウト (秒)
AGENT	ポーリングエージェントのアドレス
STATUSVAR	ポーリングステータス SNMP 変数
STATUSVAL	ポーリングステータス値
STATUSEXP	ステータス OK Expression: =, !=, <, >, <=, >=
SNMPOID	カレント sysObjectID (読み取りのみ)
NODEGROUP	ノードグループ番号
BORDER	境界色を設定(ステータスカラーの補助として使用)
HASIP	IP 対応 (読み取りのみ)
HASSNMP	SNMP 対応 (読み取りのみ)
HASTELNET	Telnet 対応
HASWEB	Web 対応
HASFTP	FTP 対応
HASRMON	RMON 対応
HASSMTP	SMTP 対応
LINKTHICKNESS	ネットワークとリンクの厚さ
LINK1	リンクコネクション数をレコードする
LINK2	他のリンクコネクション数をレコードする
POS_X	ノード横軸の位置
POS_Y	ノード縦軸の位置
GETCOMM	SNMP V1/V2 Get コミュニティ
SETCOMM	SNMP V1/V2 Set コミュニティ

TRAPCOMM	SNMP V1/V2 Trap コミュニティ
ACCESS_PARM	SNMP アクセス情報を含む特別なアクセスSTRING (V1/V2 コミュニティ名, V3 セキュリティ設定)
READ_ACCESS_MODE	読み取りアクセスモード: なし (サービスのみ), ICMP Ping, SNMP V1, SNMP V2c, SNMP V3 非認証, SNMP v3 認証-MD5, SNMP V3 認証-SHA, SNMP V3 プライベート認証-MD5, SNMP V3 プライベート認証-SHA
READ_WRITE_ACCESS_MODE	読み取り/書き込みアクセスモード: なし (サービスのみ), ICMP Ping, SNMP V1, SNMP V2c, SNMP V3 非認証, SNMP v3 認証-MD5, SNMP V3 認証-SHA, SNMP V3 プライベート認証-MD5, SNMP V3 プライベート認証-SHA
V3_ENGINEID	SNMP V3 エンジン ID
V3_CONTEXTNAME	SNMP V3 コンテキスト名
V3_READ_SECNAME	SNMP V3 読み取りセキュリティ名
V3_READ_WRITE_SECNAME	SNMP V3 読み取り/書き込みセキュリティ名
V3_AUTHPASSWD	SNMP V3 認証パスワード(8 文字以上)
V3_PRIVPASSWD	SNMP V3 プライベートパスワード(8 文字以上)
POLLSERVICES	ポーリングするサービス名のコンマ区切りのリスト

レポート設定データベースのコマンド

GRPNEXT [Acc] [-b] Recno

Recno で識別されるレポートグループオブジェクトの次のレコード番号を表示します。最初のレポートグループオブジェクトを取得するには、0 を *Recno* として使用します。

GroupName:RECNO=Recno の形式で出力されます。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-b* Bare 形式の出力。レコード番号のみ表示
- ・ *Recno* レポートグループオブジェクトのレコード番号。これ以降は next オブジェクトをかえます。最初のグループオブジェクトをかえずには 0 を使用します。

GRPADD [Acc] [-b] Name

トレンドレポート設定データベースにレポートグループオブジェクトを追加します。新しいグループオブジェクトレコード番号は *GroupName:RECNO=recno* の形式で表示されます。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-b* Bare 形式の出力。レコード番号のみを表示
- ・ *Name* レポートグループの名前

RPTNEXT [Acc] [-b] Recno [GroupRecno]

Recno で識別されるレポートオブジェクトの次のレコード番号を表示します。最初のレポートを取得するには、0 を *Recno* として使用します。 *ReportName:RECNO=Recno* の形式で出力されます。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-b* Bare 形式の出力。レコード番号のみ表示
- ・ *Recno* レポートオブジェクトのレコード番号。これ以降は *next* オブジェクトをかえします。最初のグループオブジェクトをかえすには 0 を使用します。
- ・ *GroupRecno* *GroupRecno* で指定されるグループのレポートのみをかえします

RPTADD [Acc] [-b] Group Name Table Node

トレンドレポート設定データベースにレポートオブジェクトを追加します。新しいレコード番号は *Name:RECNO=recno* の形式で表示します。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-b* Bare 形式の出力。新しいレコード番号のみ表示
- ・ *Group* このレポートを追加するグループ下のレコード番号または名前。デフォルトのレポートグループには 0 を使用します。
- ・ *Name* 新しいレポートの名前
- ・ *Table* レポートの SNMP テーブル名
- ・ *Node* レポートに追加するマップデバイスオブジェクトのレコード番号または名前

RPTATTR [Acc] [-b] Recno |Name ATTR [Value]

指定のトレンドレポートまたはレポートグループ設定レコードへまたはここから情報を書き込み/読み取ります。プログラムの出力はレポートまたはグループオブジェクト属性値です。書き込みの場合は新しい値の出力です。属性を指定せずに実行した場合は、*ReportName:Attr=Value* の形式で全ての属性の名前および値を出力します。

- ・ *Acc* SNMPc Server アクセスパラメータ
- ・ *-b* Bare 形式の出力。新しいレコード番号のみ表示
- ・ *Recno |Name* レポート設定レコードのレコード番号または名前
- ・ *ATTR* 書き込み/読み取りするレポート属性の名前
(以下の表を参照ください)
- ・ *Value* 書き込みする場合の属性値

下のテーブルでは、使用可能な属性を説明します。特別な属性形式の情報についてはこのテーブルのあとに説明します。グループオブジェクトは、RECNO、TYPE、および NAME 属性のみ有効です。

属性の名前	説明
RECNO	レポートまたはグループオブジェクトの設定データベースレコード番号
TYPE	オブジェクトタイプ: <i>Report</i> または <i>Group</i>
NAME	レポートまたはレポートグループ名
GROUP_ID	レポートオブジェクトの場合、このレポートが属するグループオブジェクトのグループ番号
NODES	レポートでポーリングを行ったマップデバイスオブジェクトの名前またはレコード番号のコンマ区切りのリスト
TABLE	レポートでポーリングした SNMP テーブル名

INSTNAMES 属性のフォーマット

INSTNAMES 属性は *Instance Name Descriptors* を大括弧('[' , ' ')で囲んだいくつものシリーズで構成されるテキスト文字列です。

各 *Instance Name descriptor* は *[Recno.Instance{Name}]* の形式で *Recno* はデバイスマップレコード番号、*Instance* はドット表記のテーブルインスタンス、*Name* にはインスタンス名を置き換えます。instance 部分には引用符で囲んだ文字列を含むことができます。

以下に *INSTNAMES* 属性値の例を記します。

[440.1{Ethernet Port 1}][440.2{Serial 1}][220.3{Dial up Port}]

[603.'Email'{Email Server}][603.'HTTP'{Web Server}]

注意: *INSTNAMES* 属性に含まれるマップレコード番号は事前に *NODES* 属性を使用してレポートに設定されている必要があります。

THRESHOLDS 属性のフォーマット

THRESHOLDS 属性は *Instance Threshold Descriptors* を大括弧('[' , ' ')で囲んだいくつものシリーズで構成されるテキスト文字列です。以下は *THRESHOLDS* 属性の例です。

[440.1{2: > 100 <= 1000}{3: > 500}][440.2{2: > 600}]

各 *Instance Threshold Descriptor* は *Recno.Instance* の形式でマップオブジェクトレコード番号およびインスタンスで始まります。次に一つ以上の中括弧({ , })で囲まれた *Variable Threshold Descriptors* が続きます。

各 *Variable Threshold Descriptor* は *Variable Table Offset* で始まりその後はコロン、*Alarm Expression*、と続きます。 *Variable Threshold Descriptor* のうちどれかひとつが True と判断された場合にアラームが生成されます。

Variable Table Offset はレポートテーブルの変数のインデックス数で 0 からはじまります。トレンドレポートには保存されず、SNMPc インスタンスダイアログで表示される文字変数は変数のテーブルオフセットに影響します。例えば、*MenuIfBPSEntry* テーブルにて *ifIndex*, *ifDescr*, *ifInBPS*, *ifOutBPS*, *ifTotalBPS* の変数があります。*ifIndex* 変数のオフセットは 0; *ifDescr* オフセットは 1 です。トレンドレポートデータでは該当がありませんが *ifInBPS* のオフセットは 2、と続きます。

Alarm Expression はいくつかのスペース区切りの表記で構成され、オペレータ

(>, <, >=, <=, =, !=) および 10 進数を含みます。オペレータの番号はスペースで区切られる必要があります。例えば、“>= 10 <= 100 != 200”。これらの表記は論理的に *OR* で表記することも考えられます。TRUE の場合には閾値アラームが生成されます。

はじめの例を *MenuIfBPSEntry* テーブルに適用した場合を以下に記します。

- Map object 440, instance “.1”, alarm if ifInBPS > 100 or ifInBPS <= 1000 or ifOutBPS > 500.
- Map object 440, instance “.2”, alarm if ifInBPS > 600.

注意: *THRESHOLDS* 属性に含まれるマップレコード番号は事前に *NODES* 属性を使用してレポートに設定されている必要があります。