

# ZABBIXによる Notes/Domino監視

株式会社ロジック

富永 泰助

# アジェンダ

- Notes/Domino監視手法検討
- NotesAPIを利用した監視
- Domino SNMPを利用した統計情報の監視
- その他Notes/Dominoで必要な監視
- 今後の展望

# Notes/Domino監視手法検討

# Notes/Dominoから見た監視すべき項目

はじめは、ZABBIXを勉強したので、Notes/Dominoの監視に適用できないか？という興味から始まりました。今回のセミナー内容はZABBIXをNotes/Domino監視に利用する方法についてテスト、運用した結果についてまとめたものです

## とりあえず立ててみた目標

サービス監視

サービス監視のテンプレートがあるので実装上支障はない

プロセス監視

プロセス監視のテンプレートがあるので実装上支障はない

ログ監視

ログ監視のテンプレートがあるので実装上支障はない

NRPCポート監視

実装方法を検討する必要がある。

統計情報の監視

実装方法を検討する必要がある。

# Notes/Dominoにおいてよくある事象

どうせやるなら次の問題をクリアした監視ソリューションとして実装してみたかった。  
Notes/Domino管理者の困りごととしては次のようなものが多い

- ▶ Domino/Dominoでよくある問題点としてネットワークの受け口は反応するが内部的に停止場合が多い
- ▶ ドミノクラスタで覆いタスクが中途半端に死んでしまっている状態
- ▶ なんとか反応しているが内部のメール滞留などで異様に遅い場合（実質停止）

つまるところ・・・

完全に落ちてくれればいいのに中途半端に落ちるのが困る

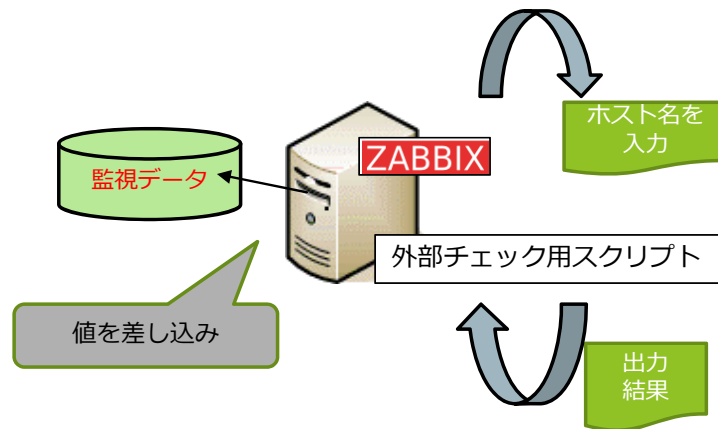
ポート監視だけではなくNotes/Dominoの内部に入り込んだ監視モジュールがあると望ましい



**ZABBIXの外部チェック機能で汎用的に値取得できるコマンドがほしい**

# ZABBIX外部チェック機能について

- ▶ ZABBIXは外部チェック機能があり外部へスクリプトを起動し値を取得することができる
- ▶ 値は数字、テキストともにZABBIXに出力を取り込むことができる
- ▶ 今回は外部チェックを使って応答時間や統計情報を出力するプログラムを実装したい

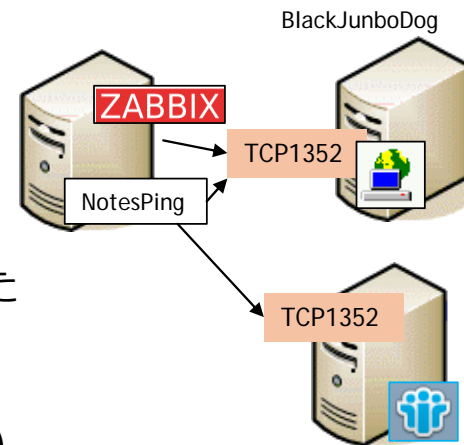


# アプローチ（１） -NotesPingの利用検討

NotesPingがチェックに利用できないか技術的な調査をおこなってみました

- ためしにZABBIXのポート監視とはなにか？  
BlackJumboDogで1352番ポートでSMTPサーバをあげてみた  
（嘘ドミノサーバー）  
⇒ 監視は通ってしまった  
やっぱりZABBIXの監視はポートのオープンしかみていない
- 同様にNotesPingでやってみた  
⇒ NotesPingでもおなじだった。  
つまりNotesPingはポートのオープンしかみていない

NotesPingの戻り値をZABBIXに解釈できるようにする等は意味がない  
（ZABBIX監視と同等）



## アプローチ（２） -通信内容の調査

次にNotes/Dominoの通信内容で何らか値がとれないか見てみました

- WWW,SMTP等でよくあるTelnetのテキストレベルで特定の文字列がかえってくるのかの動きはないか？
  - Telnet Servername 1352 を実行して確認してみた。
  - SMTP、WWW ⇒ テキストとして通信されているのでわかりやすい
  - NRPC ⇒ 通信内容が非テキストで出力されるようだ
- パケットキャプチャ結果をみてもわからない
  - ⇒ そもそもキャプチャ結果を返すスクリプトをつくっても汎用的なアクセスには使えない
  - ⇒ 専用サーバーごとにキャプチャして返すスクリプトつくれば可能か？

```
83 /6 52 b6 ...0G..b .p]b.vR.  
eb ae ed 75 ....M(+ ...$....u  
9e 5d f2 b7 ..F..a,.. ..Y!..]  
4d 41 08 00 2...)EN. ....MA..  
52 53 41 46 P.^&]... ~.PURSAF  
4f 3d 4c 4f ..CN=ZSR V01/O=LO  
GIC.
```

特定の通信などで監視を拾う方法も難しそう



# アプローチ（3） -Notes C APIでの実装

NotesクライアントをZABBIXでコントロールして通信時間を測定する

正攻法での手で考えてみました

- ZABBIXからNotesクライアントをコントロールできれば話が早い
- 動けば汎用性は高い
- C/C+ Notes APIは十数年ぶり（R4の時代！）
  - • 自分でプログラムしたのは周辺部分だけ

なんとかなるだろ！  
ということでやってみました。

## 取り出したいデータ

- ▶ names.nsf問わずDBのプロパティ取得までの時間
- ▶ EventMinitorResultでの出力を取り出す

できれば特定のビューの先頭行の特定のフィールドの値を取り出す処理にするのが望ましい)

# Notes/Domino C API を利用した監視



# ZABBIX on Notes/Domino C APIの稼働準備(1)

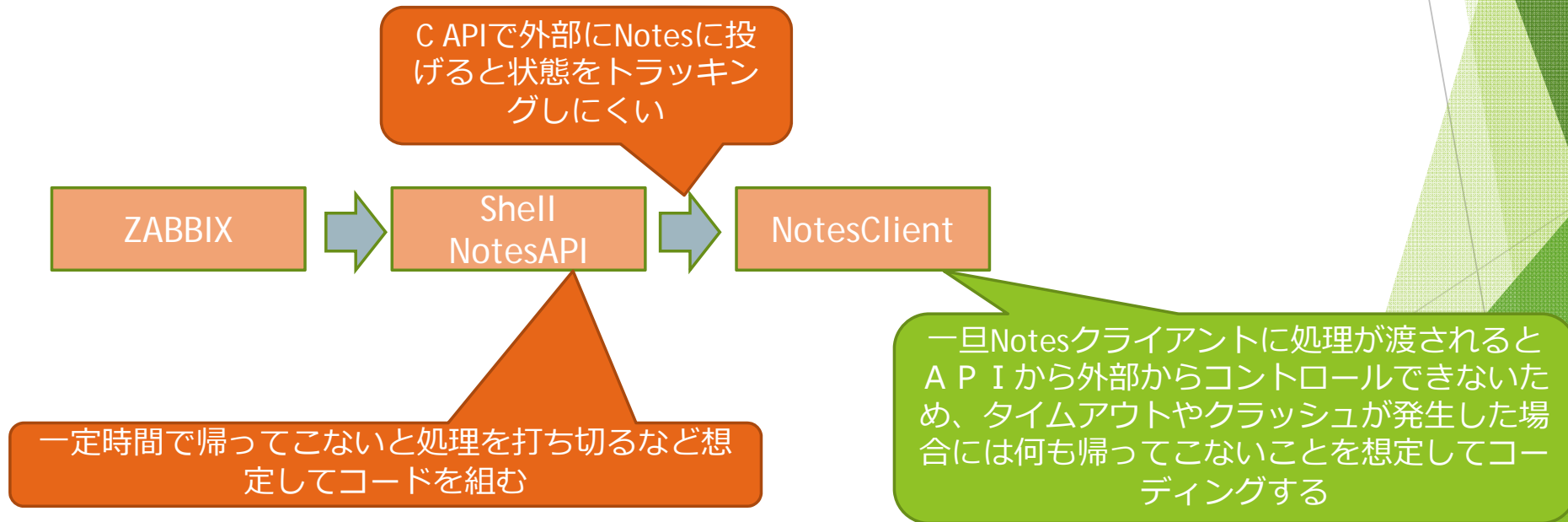
## Notes側の環境(ZABBIXインストール前)

- ▶ 先に「zabbix」アカウントをつくっておきNotesクライアントをGUIでセットアップしておく（先に/home/zabbix等を作っておく）
- ▶ GUIでKanshi/LOGIC等の監視用のID(パスワード無)でセットアップ
- ▶ ZABBIXからの外部チェックはなぜかルートフォルダからzabbixルートからアクセスされる場合ルートパスからnotes.iniを探すためエラーとなる。
- ▶ そのためリンクを作成しておく

```
In -s /home/zabbix/lotus /lotus
```

## ZABBIX on Notes/Domino C APIの稼働準備(2)

- ▶ Notes/Domino C APIアクセスするとNotesクライアントに処理が渡されるのでコードからコントロールはできない
- ▶ よって、処理を自動化する場合にはパスワード無しにして自動的に認証を通すようにする **超重要!**



# OS側のAPI稼働準備

## Notes APIのセットアップ

- ▶ CentOSで開発用ワークステーションを導入
- ▶ IBM C/C+ APIツールキットを導入

<https://www.ibm.com/developerworks/lotus/documentation/capi/>

- ▶ (環境変数にセット+特定のフォルダにAPIセットを入れる)

```
export LOTUS=/opt/ibm/lotus
export NOTES_DATA_DIR=/home/zabbix/lotus/notes/data
export Notes_ExecDirectory=/opt/ibm/lotus/notes/
export LD_LIBRARY_PATH=/opt/ibm/lotus/notes/
```

```
#cp /notesapi /opt/ibm/lotus
#chmod 755 ./notesapi
```

- ▶ アプライアンスにできたらいいなあということで、Miracle Linuxさんのアプライアンスにも実装してみました
- ▶ 50個近くのパッケージを導入する必要があるので・・・難しいです

# プログラム仕様

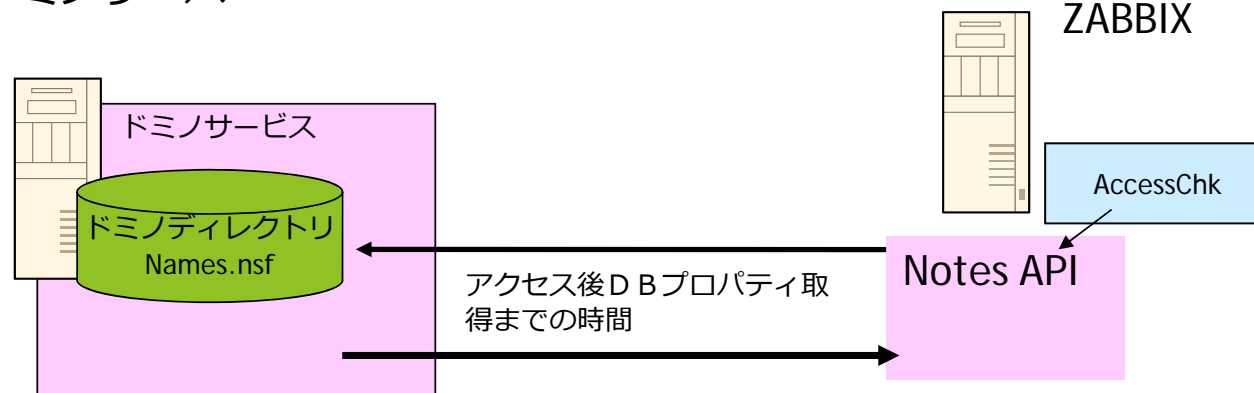
- ▶ 取り出したいデータ
  - ▶ ・ names.nsf問わずDBのプロパティ取得までの時間
  - ▶ ・ EventMinitorResultでの出力を取り出す  
(できれば特定のビューの先頭行の特定のフィールドの値を取り出す処理にするのが望ましい)
- ▶ プログラムとしてのアプローチ
  - ▶ ・ なるべく汎用的に
  - ▶ ・ なるべく簡単に
  - ▶ ・ 出力はZABBIXに合わせる

# 実装してみたアプリケーション（1）

## 1. ノーツアクセスチェック機能

NotesAPIを利用して、Notesクライアントの動きでNotes/Dominoに接続して、データベースのプロパティまでのアクセス時間をプログラム中で計測

ドミノサーバ

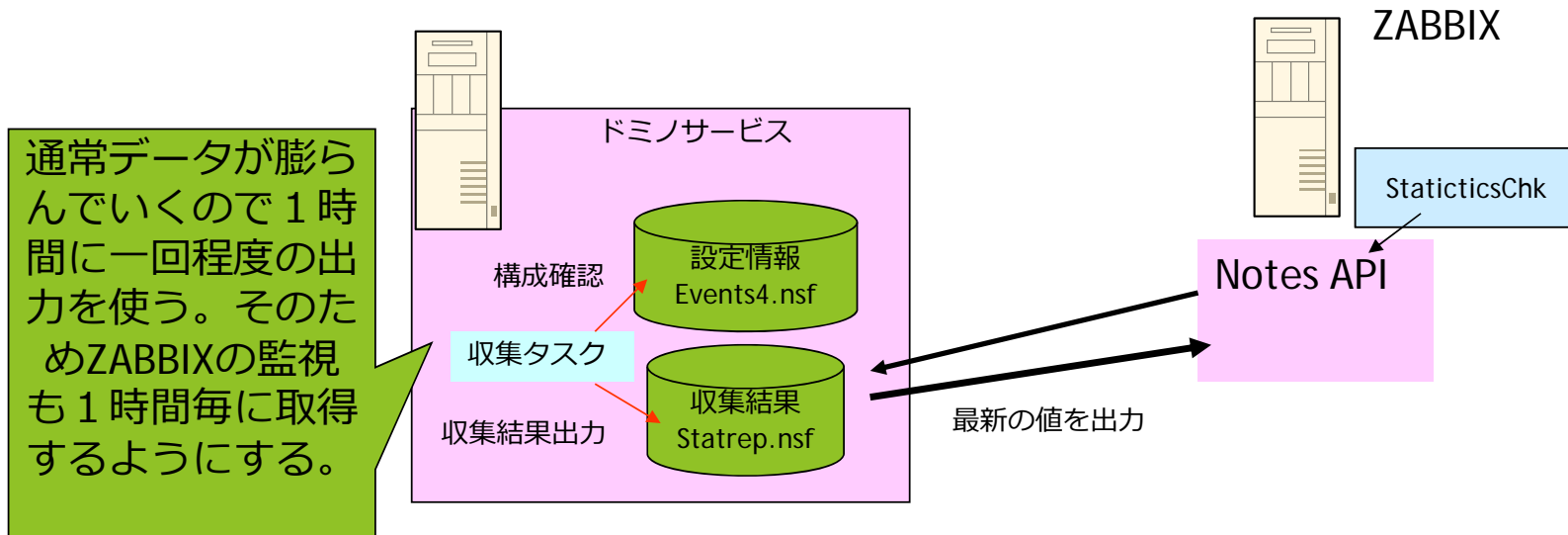


コマンド中でDB名及びサーバ名を指定できるので、他のDBのレスポンスタイム計測にも応用可能

# 実装してみたアプリケーション（2）

## 2. 統計情報アクセス機能

Lotus標準監視ソフトのイベントモニタ機能と連携して稼働するイベントモニタの収集結果をAPIアクセスにて取得する



ビューから取得するフィールド値を指定することができるので、収集結果に含まれる他のフィールド値も設定すれば取得できる



## プログラムコード（1）

- ▶ 基本はサンプルコードを参考に作成（intro.cが参考になる）
- ▶ Viewget.cの値を取り出す処理も参考になった
- ▶ ただし次の処理系を追加する
  - ・ 応答時間測定
  - ・ 戻り値をZABBIXで処理できる値に変えてあげる

```
double gettimeofday_sec(){
struct timeval t;
gettimeofday(&t,NULL);
return (double)t.tv_sec + (double)t.tv_usec * 1e-6;
}
```

## プログラムコード（2）

- ・エラー処理が考慮されていないためエラーが出た場合にはコード内で処理してZABBIXに帰してあげる（スクリプトの処理は最低限にしてシェルで受けた）

```
#!/bin/bash
export LOTUS=/opt/ibm/lotus
export NOTES_DATA_DIR=/home/zabbix/lotus/notes/data
export Notes_ExecDirectory=/opt/ibm/lotus/notes
export LD_LIBRARY_PATH=/opt/ibm/lotus/notes

retval=`/bin/bash
/usr/lib/zabbix/externalscripts/getnotesstat.sh $1
DataBase.DbCache.Hits`

if [ -x $retval ]
then
echo 99999

else
echo $retval

fi
```

一本で行いたかったが2本のプログラムになった

- ・DBプロパティ取得時間出力プログラム
- ・ビュー値出力プログラム

# ZABBIXでの外部チェックの設定

特定のパスにおいて、ZABBIXから設定します。

/usr/lib/zabbix/externalscripts  
にzabbixアカウントで実行権限を  
つけておく

ホスト	ZSRV01						
名前	NotesCheck2						
タイプ	外部チェック						
キー	noteschk.sh[HOST.IP]						
ホストインターフェース	192.168.1.2 : 10050						
データ型	数値 (整数)						
データの形式	10進数						
単位	msec						
乗数を使用	<input type="checkbox"/> 1						
更新間隔 (秒)	30						
例外の更新間隔	<table border="1"><thead><tr><th>更新間隔</th><th>期間</th><th>アクション</th></tr></thead><tbody><tr><td colspan="3">例外の更新間隔は設定されていません。</td></tr></tbody></table>	更新間隔	期間	アクション	例外の更新間隔は設定されていません。		
更新間隔	期間	アクション					
例外の更新間隔は設定されていません。							
例外の更新間隔の作成	更新間隔 (秒) 50 期間 1-7,00:00-24:00 追加						
履歴の保存期間 (日)	90						
トレンドの保存期間 (日)	365						
保存時の計算	なし						
値のマッピングの使用	なし <a href="#">値のマッピングの表示</a>						
アプリケーションの作成							
アプリケーション	なし						

```
[root@zabbixtest externalscripts]# ls -al
-rwxr-xr-x 1 zabbix zabbix 338 4月 4 2013 getdbcachehit.sh
-rwxr-xr-x 1 zabbix zabbix 326 4月 4 2013 getdeadmail.sh
-rwxr-xr-x 1 zabbix zabbix 9374 7月 19 2012 getnotesstat
-rwxr-xr-x 1 zabbix zabbix 271 4月 4 2013 getnotesstat.sh
-rwxr-xr-x 1 zabbix zabbix 329 4月 4 2013 getpendingmail.sh
-rwxr-xr-x 1 zabbix zabbix 339 4月 4 2013 gettransaction.sh
-rwxrwxrwx 1 zabbix zabbix 9380 7月 19 2012 notes_check
-rwxrwxr-x 1 zabbix zabbix 9380 7月 19 2012 notes_chk
-rwxr-xr-x 1 zabbix zabbix 245 4月 4 2013 noteschk.sh
[root@zabbixtest externalscripts]#
```

```
[root@zabbixtest externalscripts]# cat noteschk.sh
#!/bin/bash
export LOTUS=/opt/ibm/lotus
export NOTES_DATA_DIR=/home/zabbix/lotus/notes/data
export Notes_ExecDirectory=/opt/ibm/lotus/notes
export LD_LIBRARY_PATH=/opt/ibm/lotus/notes

/usr/lib/zabbix/externalscripts/notes_check $1 names.nsf
```

中身は単純なスクリプト

## これで完成！のはずが・・・

- ▶ 実装が無事完了したので50台を想定して同一サーバーにアクセスする形をつくったところ・・・
- ▶ NotesAPIを多重起動するとNotesAPIがタイムアウトしてZABBIX外部スクリプトからタイムアウトしてしまう。
- ▶ 試していたところ、Core2Duo/RAM3Gで5分間隔で20ぐらいは問題ないが、それ以上だとかなりの確率でエラーになってしまう

解決策：どうもMonitoring Resultからの出力が重いことがわかった  
値を一つ取り出すのにNotesAPI開始⇒値取り出し⇒NotesAPI終了  
まで行っている

Notesの標準的な監視項目（最少でも数個から多いと20個）＊サーバー台数分＝となると実装は厳しい

## 解決案の模索

- ▶ Events出力の取り出しはSNMPにまかせてAPIでしかできない内部実応答時間だけAPIにまかせるアプローチ
- ▶ これならサーバー台数分なので、一定時間ごとにサーバー台数分なので大した負荷ではない
- ▶ Notes応答時間は処理の中に処理時間をずらす処理をいれてNotesAPI多重実行を抑止
- ▶ これによりエラー出力はなくなった

```
srand((unsigned) time(NULL));  
sleeptime = rand() / 40000;  
usleep(sleeptime);
```

統計情報など負荷が重い処理に関しては  
SNMPを利用して引き出す方向を模索

# Domino SNMPを利用した監視

# Domino SNMPの利用(1)

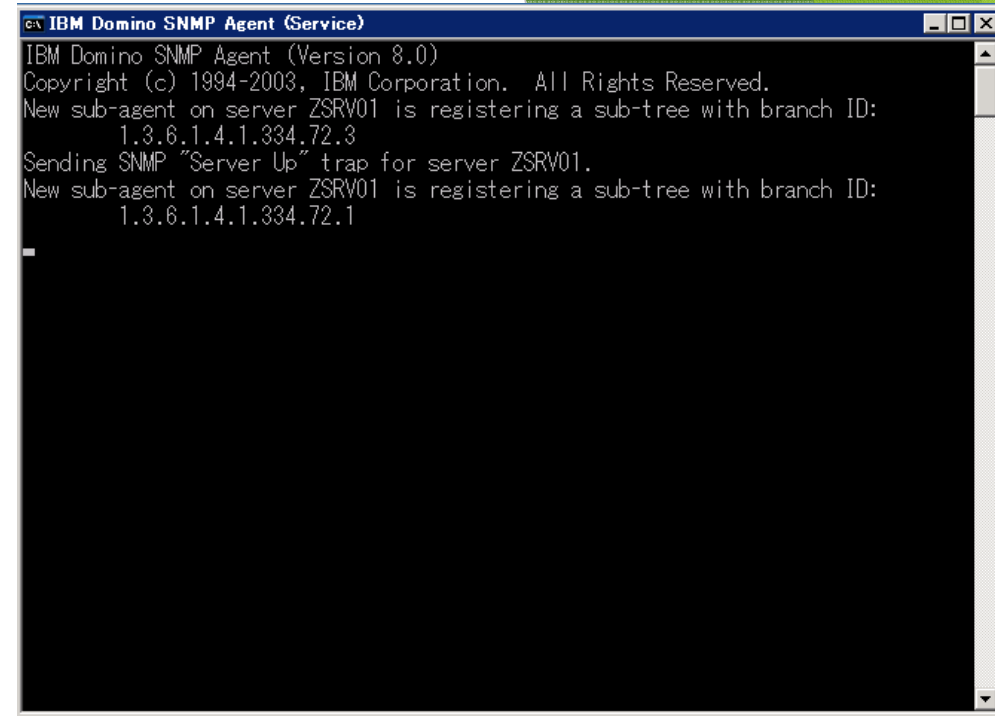
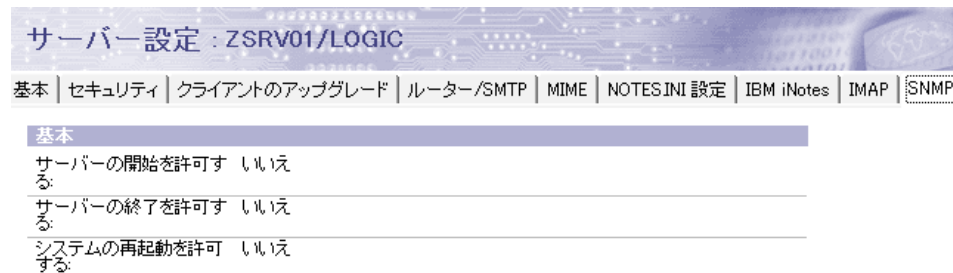
## Domino SNMPの利用

前提条件としてWindows SNMP等が有効になっている必要  
その上で、DominoSNMPサービスを組み込みます  
このサービスは⇒のように別サービスで常駐します  
#Insnmp -Sc

次にDominoサーバー側で必要なサービスを動かします  
またServerTasks組み込みます

load collect (統計情報を取り出すのに必要)  
load qryset (外部からの照会を受け付けるのに必要)  
load intrcpt (外部からのコントロールを受け付けるのに必要)

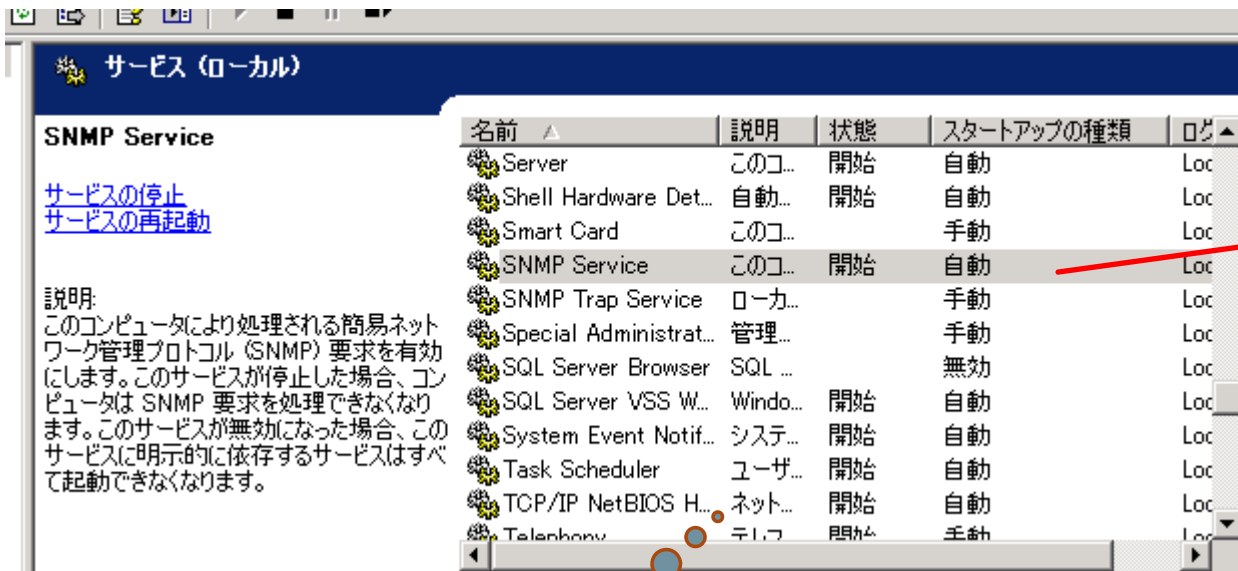
さらにサーバー設定文書で外部からのSNMPトラップで再起動、停止、起動を許可するかを設定します



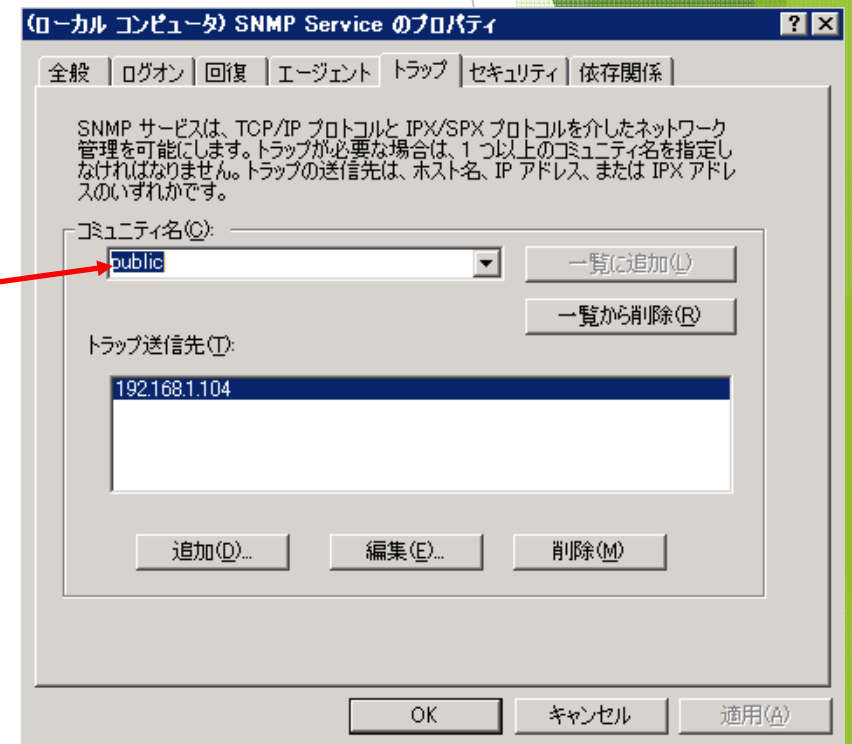
# Domino SNMPの利用(2)

## OS側の設定

次にOSの設定を行います。DominoSNMPを有効にすると、Dominoサーバというよりもネットワークアプライアンス機器のような扱いになるので、SNMPトラップ、参照はOS側が管理します



準備完了



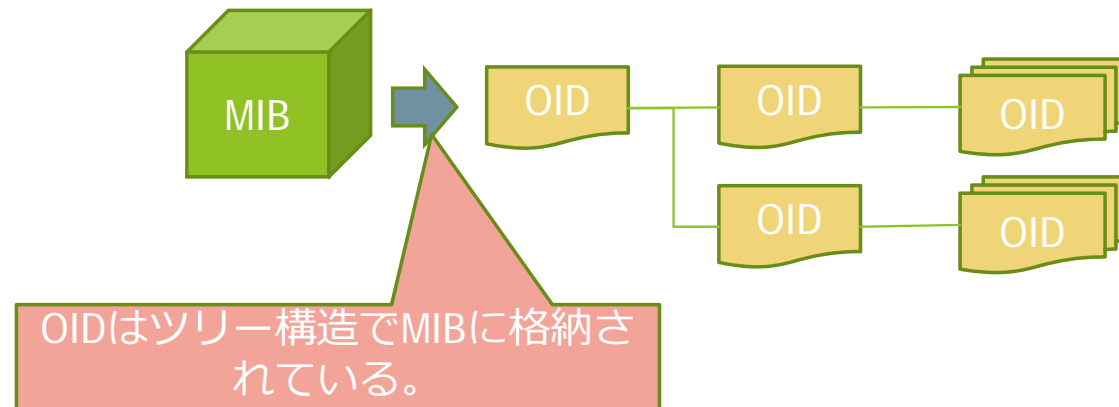


# DominoSNMPの調査(1)

## Domino SNMP MIB調査

まずはMIBを調査してDominoサーバーがどのような機能をもっているのか調べます

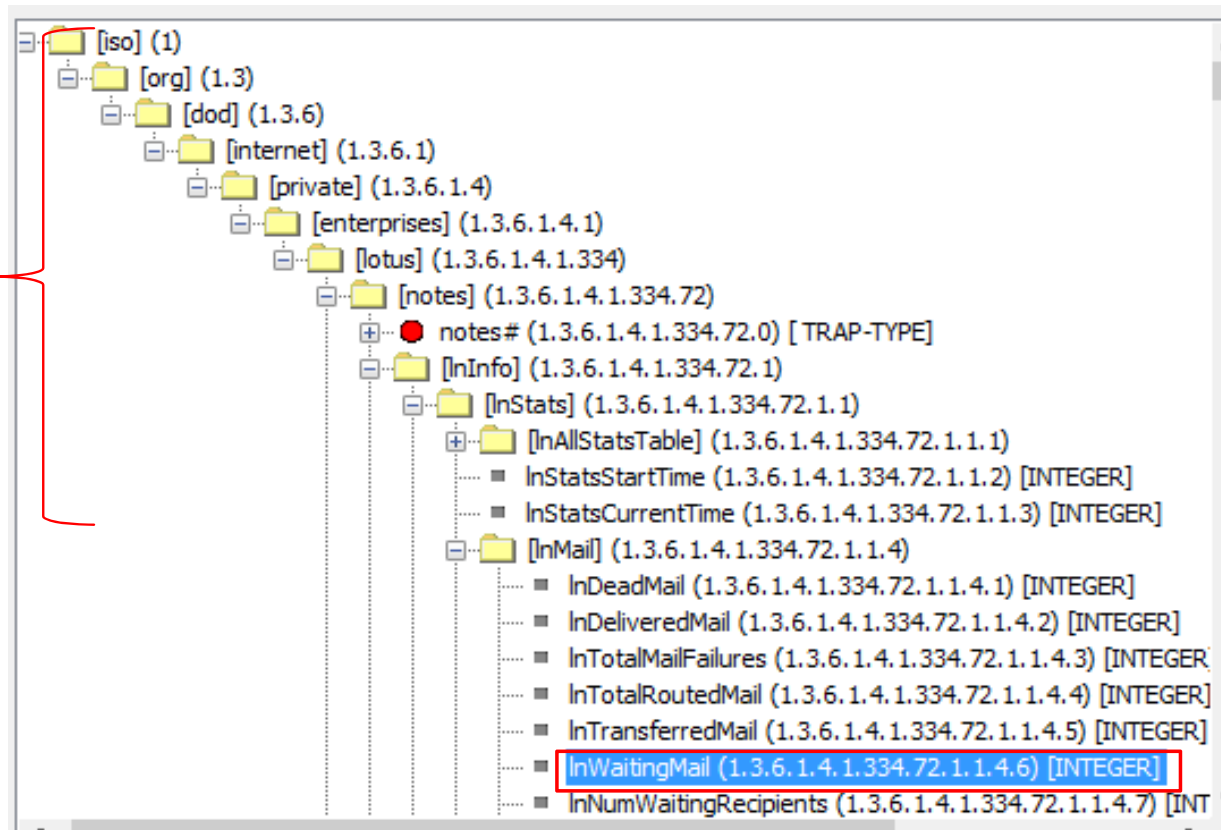
MIB(Management information base)とはDomino（というネットワーク機器）がどういった監視項目をもっているかのリスト。DominoサーバーのプログラムフォルダのDomino.MIBがそれにあたる。これをMIBブラウザというプログラムで読み込んで監視内容(Object Identifier)を取得できる。SNMPからは特定のOID番号を読めば監視情報を取り出すことができる



# DominoSNMPの調査(2)

Windows用のMIBブラウザは少ないですが私はTWSNMPマネージャを使っています。下記は、MIBブラウザのサンプルです

MIBはツリー構造になっています。滞留メール数は「iso/org/dod/internet/private/enterprise/lotus/notes/InInfo/InStats/InMain」の中にOIDがあることがわかります。



InWaitingMailで値をもっており[INTERGER]の整数値を持っていることがわかります。またOIDは1.3.6.1.4.1.334.72.1.1.4.6であることがわかります。

# DominoSNMPの調査(3)

## MIB説明の参照

MIBにはオブジェクト情報と一緒にその値に対する説明もあります

```
InWaitingMail OBJECT-TYPE
-- FROM NOTES-MIB
SYNTAX INTEGER (0..2147483647)
MAX-ACCESS      read-only
STATUS optional
DESCRIPTION     "Number of mail messages waiting to be routed."
 ::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) lotus(334) notes(72) InInfo(1)
       InStats(1) InMail(4) 6 }
```

この値が取り得る値および許可されているアクセス権（ここでは読み込み専用）および説明が記載されています

# DominoSNMPの調査(4)

## SNMPトラップ

SNMPは読出しだけでなくDominoが落ちたら特定のSNMPトラップを出力することや、外部から特定のSNMP値を受け付けるとサーバーを停止、起動する機能もあります

```
InServerTblSetState OBJECT-TYPE
-- FROM NOTES-MIB
SYNTAX INTEGER {up(1), down(2)}
MAX-ACCESS      read-write
STATUS optional
DESCRIPTION     "This will indicate the current state of the Domino server and
allow remote SNMP control over stopping & starting the Domino
server. The server will be considered down until at least one
of the Domino management Agent addin tasks starts and registers
with the Main Proxy Agent."
 ::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) lotus(334) notes(72) InControl(2)
InServerTable(6) InServerEntry(1) 8 }
```

この値はRead-writeが許可されとりこの値にUp、Downをかくことよりコントロールされることを示します

# ZABBIXでのDominoSNMP監視(1)

では実際にZABBIXでDomino SNMPを読み込んでみました。

ホスト	ZSRV01						
名前	Mail.Waiting						
タイプ	SNMPv2エージェント						
キー	Mail.Waiting						
ホストインターフェース	192.168.1.2 : 161						
SNMP OID	1.3.6.1.4.1.334.72.1.1.4.6.0						
SNMPコミュニティ	public						
ポート							
データ型	数値 (整数)						
データの形式	10進数						
単位							
乗数を使用	<input type="checkbox"/> 1						
更新間隔(秒)	30						
例外的更新間隔	<table><thead><tr><th>更新間隔</th><th>期間</th><th>アクション</th></tr></thead><tbody><tr><td colspan="3">例外的更新間隔は設定されていません。</td></tr></tbody></table>	更新間隔	期間	アクション	例外的更新間隔は設定されていません。		
更新間隔	期間	アクション					
例外的更新間隔は設定されていません。							
例外的更新間隔の作成	更新間隔 (秒) 50 期間 1-7,00:00-24:00 追加						
履歴の保存期間(日)	90						
トレンドの保存期間(日)	365						
保存時の計算	なし						
値のマッピングの使用	なし <a href="#">値のマッピングの表示</a>						
アプリケーションの作成							

キーはわかりやすい識別であれば何でもかまいません

OIDは「1.3.6.1.4.1.334.72.1.1.4.6」となります。さらに最初の値をもってくる意味で末尾に「.0」を追加します。

# 主要なOID

ZABBIXで下記のOIDを監視設定してみました。

統計名	OID	収集名
InWaitingMail	1.3.6.1.4.1.334.72.1.1.4.6	メール滞留数
InDeadMail	1.3.6.1.4.1.334.72.1.1.4.1	デッドメール数
InMailTransferFailures	1.3.6.1.4.1.334.72.1.1.4.19	メール配送失敗数
InRepFailed	1.3.6.1.4.1.334.72.1.1.5.4	複製失敗数
InRepClusterFailed	1.3.6.1.4.1.334.72.1.1.5.6.4	クラスタ複製失敗数
InServerTransPerMin	1.3.6.1.4.1.334.72.1.1.6.3.2	1分間当たりのトランザクション数
InServerUsers	1.3.6.1.4.1.334.72.1.1.6.3.6	サーバーへの接続ユーザー数

値が出力されないものは取得不可になります。たとえばクラスタ化されていないサーバや複製の失敗が発生していないサーバ等は取得不可になります。

よってサーバーによっては取得アイテムの調整及び取得不可になった場合の再取得間隔の調整をおこないます。このZABBIXでは10分として設定しています。この時にはレプリカの失敗から最大11分程度で通知されます

名前 ↑	最新のチェック	最新の値	変更
- other - (9アイテム)			
agentping	2013/11/30 14:54:55	1	
Domino HTTP Access	2013/11/30 14:53:52	1 msec	
Domino Log Check	2013/11/30 14:54:42	[ODFC:003F-08F8] 201 ...	
InDeadMail	2013/11/30 14:54:41	7 Mail	-
InRepFailed	2013/11/30 14:54:43	7	-
InServerTransPerMin	2013/11/30 14:54:45	11	-3 <a href="#">グラフ</a>
InServerUsers	2013/11/30 14:54:46	2	- <a href="#">グラフ</a>
Mail.Waiting	2013/11/30 14:54:40	0	- <a href="#">グラフ</a>
NotesCheck2	2013/11/30 14:54:26	729 msec	+32 msec <a href="#">グラフ</a>

# リモートでのDominoコントロール (1)

次にアラートを発生した場合、サーバーを再起動させるように設定してみましょう。  
最初にO I Dを調査します。

The screenshot shows a tree view of network objects. The 'InControl' folder is expanded, and 'InNotesServerSetState' is selected and highlighted with a red box. A callout box provides the following details for this object type:

```
InNotesServerSetState OBJECT-TYPE
-- FROM NOTES-MIB
SYNTAX INTEGER {up(1), down(2)}
MAX-ACCESS read-write
STATUS mandatory
DESCRIPTION "Set to 'down' to shut Notes server down, set to 'up' to
restart Notes server. Note: access to this feature is
controlled through external (non-SNMP) security measures."
 ::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) lotus(334) notes(72) InControl(2)
1 }
```

この値をみると、書込みが許可されており、かつ起動、停止の状態をもっているようです。ここに2がはいれば、サーバーが停止していると確認できます

## リモートでのDominoコントロール (2)

- ▶ サーバー停止用のスクリプトを作ります
- ▶ snmpのコマンドで停止用MIB書込みを行います
- ▶ 設定は、先ほどつくったmail.Waitingが1以上の場合、先ほど調べたSNMP OID(InServerTblState(1.3.6.1.4.1.334.72.2.6.1.8))にdown(2)をいれればよいことがわかります。
- ▶ さらに汎用的にIPアドレス指定できるようにします。
- ▶ パスは/usr/lib/zabbix/alertscriptsにつくりzabbixユーザーで実行権限を足しましょう。

```
#!/bin/sh  
snmpset -v2c -c "public" $1 "1.3.6.1.4.1.334.72.2.1.0" "i" "2"
```



## リモートでのDominoコントロール (3)

次にアラートを発生させ、サーバーを停止させるように設定します。アクションでリモートコマンドを作成します。リモートコマンドはZABBIXサーバー上で稼働させますので特にAgentdのリモートコマンドを有効にする必要はありません。

### トリガー

名前	Mail Waiting Alert
条件式	{ZSRV01:Mail.Waiting.prev(0)}>0
条件式ビルダー	<a href="#">条件式ビルダー</a>
説明	MailWaitingAlert
URL	
深刻度	<input type="checkbox"/> 未分類 <input type="checkbox"/> 情報 <input type="checkbox"/> 警告 <input type="checkbox"/> 軽度の障害 <input type="checkbox"/> 重度の障害 <input type="checkbox"/> 致命的な障害
有効	<input checked="" type="checkbox"/>

ZABBIXサーバー上で動かします

### アクション

ステップ	開始	1
	終了	1 (0 - 無限)
	ステップの期間	0 (最小:60秒, 0 - デフォルト設定を利用)
実行内容のタイプ	リモートコマンド	
ターゲットリスト	ターゲット	アクション
	現在のホスト	削除
	新規	
タイプ	カスタムスクリプト	
次で実行	<input type="radio"/> Zabbixエージェント <input checked="" type="radio"/> Zabbixサーバー	
コマンド	/usr/lib/zabbix/alertscripts/serverstop.sh {IPADDRESS}	
アクションの実行条件	ラベル	名前
	新規	
更新 キャンセル		

スクリプトはフルパスで記載します。後ろにはマクロでIPアドレスを設定します。

## リモートでのDominoコントロール (4)

サーバーが停止された場合サーバーダウントラップを発行してサーバーが停止されます。下記は何回か起動、再起動を繰り返していますがDominoSNMPマネージャのサンプル画面です

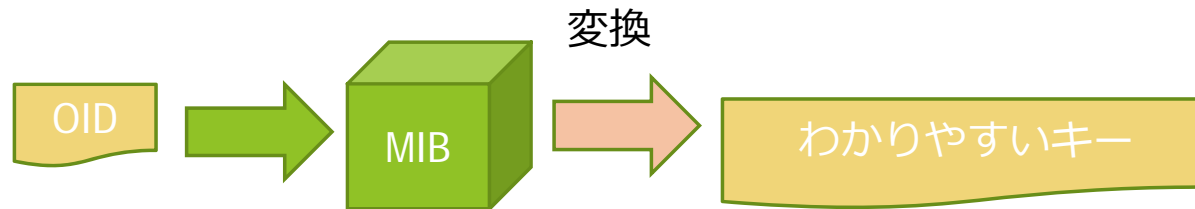
```
IBM Domino SNMP Agent (Service)
1.3.6.1.4.1.334.72.1
Sending SNMP "Server Up" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sub-agent on server ZSRV01 is unregistering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.1
Sub-agent on server ZSRV01 is unregistering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sending SNMP "Server Down" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sending SNMP "Server Up" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.1
Sub-agent on server ZSRV01 is unregistering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sub-agent on server ZSRV01 is unregistering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.1
Sending SNMP "Server Down" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sending SNMP "Server Up" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.1
```

サーバーを停止されるとトラップを発行しています

またサーバー起動時にもトラップを発行しているのがわかります

## プライベートMIBの設定（1）

- ▶ 最後にZABBIX側にDominoプライベートMIBを読み込んでわかりやすくしましょう
- ▶ プライベートMIBを読み込むことによりOIDの数字ではなく識別しやすいキーワードでOID名を設定することができます。



```
[root@zabbixtest zabbix]# snmptranslate .1.3.6.1.4.1.334.72.1.1.4.6
NOTES-MIB::InWaitingMail
[root@zabbixtest zabbix]# snmptranslate .1.3.6.1.4.1.334.72.1.1.4.1
NOTES-MIB::InDeadMail
[root@zabbixtest zabbix]#
```

## プライベートMIBの設定（2）

- ▶ プライベートMIBはZABBIXユーザーに対して設定します
- ▶ 最初に/home/zabbix/.snmpフォルダを作成します
- ▶ さらにmibsフォルダを作成し、その中にdomino.mibをコピーします。
- ▶ /home/zabbix/.snmpフォルダにsnmp.confを下記内容で作成します。

```
MIBDIRS /usr/share/snmp/mibs:/home/zabbix/.snmp/mibs  
MIBS all
```

## プライベートMIBの設定（3）

ホスト	ZSRV01						
名前	InDeadMail_SNMP_MIB						
タイプ	SNMPv2エージェント						
キー	InDeadMail_SNMP_MIB <input type="button" value="選択"/>						
ホストインターフェース	192.168.1.2 : 161						
SNMP OID	NOTES-MIB::InDeadMail.0						
SNMPコミュニティ	public						
ポート							
データ型	数値（整数）						
データの形式	10進数						
単位	Mail						
乗数を使用	<input type="checkbox"/> 1						
更新間隔（秒）	30						
例外の更新間隔	<table><thead><tr><th>更新間隔</th><th>期間</th><th>アクション</th></tr></thead><tbody><tr><td colspan="3">例外の更新間隔は設定されていません。</td></tr></tbody></table>	更新間隔	期間	アクション	例外の更新間隔は設定されていません。		
更新間隔	期間	アクション					
例外の更新間隔は設定されていません。							

単純にsnmptranslateあるいはMIBブラウザで見つかったキーを入れます。

たとえばDeadMailなら  
NOTES/MIB::InDeadMail.0等にします。

その他 Notes/Dominoで必要な監視

# ログの監視（1）

ログの監視は、コンソールログをテキスト監視します。ちょうどDominoサーバーはIBM\_TECHNICAL\_SUPPORTフォルダの下にconsole.logを常に最新版を出力しますので、ここで設定します

ただしDebug\_OutFile等で出力場所を変えている場合にはそれに合わせて設定します

設定上はすべてのログファイルをZABBIXに取り込んだうえで扱うことも可能なのですが、大変な量になるので気になるところだけテキストで出力するようにします。次のようなログを拾うように設定しました

監視するログ	よくある現象
破損	メールファイルが破損して配送ができない場合にこのエラーがでます。
error /Error	とりあえずエラーが出たらすべて記録としておきましょう
READ ONLY	メールファイルやDBファイルが破損すると読み取り専用になります
panic	ログでDominoサーバーがクラッシュすると最後に出ることがあります。

他は使っている環境にもよりますが、D A O Sを使っている環境であればD A O Sの不整合を示すNEED RESYNCやDAOSMGR関連から拾っておいたほうがいいかとおもいます。またセキュリティの違反を示すInternetpassword lockout関連のエラーやウイルス関連のエラーも出力させてログ経由で拾うこともできます。

# ログの監視 (2)

ZABBIXの設定は下記のようにします。ログの検索は正規表現出かれますので必要におうじて変更します。

The screenshot shows the Zabbix configuration interface for a log check item. The fields are as follows:

- ホスト: ZSRV01
- 名前: Domino Log Check
- タイプ: Zabbixエージェント(アクティブ)
- キー: log[C:\Lotus\Domino\Data\IBM\_TECHNICAL\_SUPPORT\c (highlighted with a red box)
- データ型: ログ
- 更新間隔(秒): 1
- 履歴の保存期間(日): 90
- ログの時間の形式: (empty)
- アプリケーションの作成: (empty)
- アプリケーション: なし
- 説明: (empty)
- ステータス: 有効

Buttons at the bottom: 保存, 複製, 履歴とトレンドを削除, 削除, キャンセル

log[C:¥Lotus¥Domino¥Data¥IBM\_TECHNICAL\_SUPPORT¥console.log,(panic|corrupt|READ ONLY|破損|Error|error),SHIFT\_JIS]

ログはすべて拾わずに正規表現で必要な分だけ拾うようにします。(文字列1|文字列2|文字列3)のように記載します。



# ポートの監視

ポートの監視も設定しておきます。主要な通信ポート、WWW(TCP 80),NRPC(TCP 1352),SMTP(TCP 25)を設定します。以下のように設定します。

ホスト	ZSRV01						
名前	Domino HTTP Access						
タイプ	シンプルチェック						
キー	net.tcp.service[http,,80] <input type="button" value="選択"/>						
ホスト-インターフェース	192.168.1.2 : 10050						
データ型	数値 (整数)						
データの形式	10進数						
単位	msec						
乗数を使用	<input type="checkbox"/> 1						
更新間隔(秒)	500						
例外の更新間隔	<table><thead><tr><th>更新間隔</th><th>期間</th><th>アクション</th></tr></thead><tbody><tr><td colspan="3">例外の更新間隔は設定されていません。</td></tr></tbody></table>	更新間隔	期間	アクション	例外の更新間隔は設定されていません。		
更新間隔	期間	アクション					
例外の更新間隔は設定されていません。							
例外の更新間隔の作成	更新間隔 (秒) 50 期間 1-7,00:00-24:00 <input type="button" value="追加"/>						
履歴の保存期間(日)	90						
トレンドの保存期間(日)	365						
保存時の計算	なし						
値のマッピングの使用	なし <input type="button" value="値のマッピングの表示"/>						
アプリケーションの作成							

Net.tcp.service[http,,80]などZABBIXでの標準的なキーを設定します。

# 今後の展望



# ZABBIXでNotes監視のメリット

- ▶ アラートを上げないでも値の履歴をもっているところ。特にトランザクション数の遷移等をグラフでもっておけます
- ▶ 特にログは条件に引っかかるものだけをあつめておくことができる。これによりサーバー障害のときにログを追っかけるときに古いログが1か月前からずーっと問題がおきていたのか等をおうことができる。NSFログファイルは標準では7日でけされることや、テキストログも場合によってははけしているため障害の要因が前から発生していた履歴を追える
- ▶ さらにサーバー台数がおおいとログをみるのが大変だが、ZABBIXでサマリログを取得することができます。よって、日常運用はログはサマリログだけで調査でサーバーログをみるなどの使い方をすることができます
- ▶ 他システムと同列にDominoの稼働状況も一覧でみることもできる
- ▶ ・ DominoAdministratorでは一覧表示も可能だが、ZABBIXではほかのシステムも合わせてアラート監視画面に一覧表示できる

# 今後の実装

このセミナーではまにあいませんでしたでしたが、ZABBIX 2.0からSNMPトラップのサポートが改善されているようです。ただしサポートについては追加のモジュール (snmptt)が必要です。がんばってはみたのですが、動きませんでした。おそらくテストしているOSがCentOS5ですのでモジュールが古いと思っています。

これが実装できればサーバー自身でDominoサービスを監視してUp,Downのときにアラートをだすこと可能です。タスク自身とDominoSNMPサービスは30秒間隔で通信しているため内部に入り込んでいるため精度の高い監視が行えると思っています。今後はAPI部分を極力SNMP等の実装に切り替えていきたいと思っています。これであればZABBIX SIAさんのアプライアンスにも組み込みやすいと思っています。

良い情報があればご教示ください  
m(\_ \_)m

```
IBM Domino SNMP Agent (Service)
IBM Domino SNMP Agent (Version 8.0)
Copyright (c) 1994-2003, IBM Corporation. All Rights Reserved.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.1
Sending SNMP "Server Up" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sub-agent on server ZSRV01 is unregistering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.1
Sub-agent on server ZSRV01 is unregistering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sending SNMP "Server Down" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.3
Sending SNMP "Server Up" trap for server ZSRV01.
New sub-agent on server ZSRV01 is registering a sub-tree with branch ID:
1.3.6.1.4.1.334.72.1
```

# 最後に . . . .

- ▶ ZABBIXはNotes/Dominoの監視を広げてくれます
- ▶ Domino SNMPとZABBIXは相性がいいと思っています

何かあればお気軽に連絡ください！

株式会社ロジック 富永 泰助  
Mail [t-tominaga@logic-inc.co.jp](mailto:t-tominaga@logic-inc.co.jp)

ご清聴ありがとうございました！