

本TIPsは、以下の場面で活用します。

リスナー経由でDB接続の異常を素早く検知し通知する

※ システム稼働状況によって本手順の適切値と運用影響が異なりますので、本番環境への適用時には、事前に十分な検証を実施してください。

DB接続成否の監視スクリプト(以下「db\_connection\_check.sh」)を「~/script」に配置します。※スクリプトの詳細については、補足情報のページをご参照ください。

## ■「db\_connection\_check.sh」(1/2)

```
$ vi db_connection_check.sh

#!/bin/sh
#*****
# ■Usage Sample
#   ./db_connection_check.sh DBNAME TNS_NAME 0
# ■検証済みOS
#   Linux
#*****

script_name=`basename $0`
process_cnt=`ps -ef | grep ${script_name} | grep ${1} | grep -v grep | grep -v 'sh -c' | grep -v ${2} | wc -l`
if [ ${process_cnt} -ne 0 ]; then
    echo "Same process exists!! Only one process running"
    echo 0; exit;
fi

db_name=$1
tns_name=$2
if [ $# -eq 3 ]; then
    rotate_month=$3
else
    rotate_month=0
fi
CONF_NAME=`echo ${db_name} | tr "a-z" "A-Z"`
ALERTLOG_DIR=${MXG_HOME}/conf/${CONF_NAME}/ALERTLOG
ALERTLOG_FILE=ALERT`date '+%y%m%d'`.log
HISTORY_LOG=${db_name}_connection_`date '+%Y%m'`.log
DBCONN_STR="maxgauge/maxgauge@${tns_name}"
INDICATOR_NAME="${db_name} connection"
```

## ■「db\_connection\_check.sh」(2/2)

```
SQL_STRING="`echo ${SQL_STRING}` SET HEADING OFF ECHO OFF PAGESIZE 0 TRIMSPOOL ON FEEDBACK OFF TIMING OFF"
SQL_STRING="`echo ${SQL_STRING}` ¥nalter session set nls_date_format = 'HH24:MI:SS';"
SQL_STRING="`echo ${SQL_STRING}` ¥nselect sysdate||' '|| status from v¥$instance;¥n"

DBCONN_REQUEST_TIME=`date '+%Y-%m-%d %T'`
SQL_RESULT=`echo -en ${SQL_STRING} | ${ORACLE_HOME}/bin/sqlplus -s ${DBCONN_STR}`

dbconn_response_time=`echo ${SQL_RESULT} | awk '{print $1}'`
indicator_var=`echo ${SQL_RESULT} | awk '{print $NF}'`
if [ ${indicator_var} != 'OPEN' ]; then
    dbconn_response_time=`date '+%H:%M:%S'`
    indicator_var="ERROR"
    echo ${dbconn_response_time}$'¥t'${INDICATOR_NAME}$'¥t'${indicator_var}$'¥t'2 >> ${ALERTLOG_DIR}/${ALERTLOG_FILE}
    echo 1;
else
    echo 0;
fi
if [ ${rotate_month} -ne 0 ]; then
    echo ${DBCONN_REQUEST_TIME}', '${dbconn_response_time}', '${indicator_var}' >> ${ALERTLOG_DIR}/${HISTORY_LOG}
fi

delete_month=`date -d "${rotate_month} month ago" +%Y%m`
cd ${ALERTLOG_DIR}
for file in ${db_name}_connection_*.log
do
    file_month=`expr ${file} : "${db_name}_connection_¥([0-9]¥{6}¥).log"`
    if [ -n "$file_month" ] && [ ${file_month} -le ${delete_month} ]; then
        rm -f ${ALERTLOG_DIR}/${file}
    fi
done
sleep 1

$ chmod 700 db_connection_check.sh
```

# 活用TIPs:DB接続成否を監視する(2/2)

\$MXG\_HOME/conf/<CONF\_NAME>/alert.conf ファイルへ監視設定を追加します。

## □ 概要

DB接続成否(db\_connection\_check.sh)の監視スクリプトを指定間隔で実行し、戻り値が‘0’でない場合 R.T.M に指定文字列のアラートを通知する

## □ 構文

Script<n>=<アラートの文字列>、<スクリプトの実行間隔(秒)>、<正常の戻り値>、<スクリプトファイルのフルパス> <DB名> <TNS接続識別子> <履歴の保存期間(月単位)>

## <推奨設定>

指標	実行間隔(秒)	履歴の保存期間(月)
DB接続(DBサーバー上)	60	0

設定例) 「orcl」DB へ「ORCL」接続経由の接続情報を1分おきに試し接続できない場合アラートをあげる

```
$ cd $MXG_HOME/conf/ORCL
$ vi alert.conf

script1=db connection alert ,60,0,~/script/db_connection_check.sh orcl ORCL 0

$ mxgctl stop ORCL RTAD
$ mxgctl start ORCL RTAD
```

 引き続き、メール設定を行う場合は MaxGauge TIPs「[DBの死活監視を行う](#)」の4ページからご参照ください。

## □ シェルの概要

- ・ DB接続成否を監視し、接続できない場合アラートを上げる。
- ・ スクリプト実行時に、DB接続成否の情報を履歴として記録する。

## □ 引数

1. DB名
2. TNS接続識別子
3. 履歴の保存期間(月単位)

## □ 戻り値

‘0’ (正常)    ‘1’ (異常)

※異常の場合、MaxGauge アラートロ( `$MXG_HOME/conf/<CONF_NAME>/ALTERLOG/alertYYMMDD.log` )  
にアラートの内容を記録する

## □ スクリプトの構成

- ・ シェル名: `db_connection_check.sh`
- ・ 履歴ファイル

– `$MXG_HOME/conf/<CONF_NAME>/ALTERLOG/<DB名>_connection_YYYYMM.log`

記録フォーマット: 接続要求時間、接続確立時間、DBのステータス(`v$instance.status` 値 OPEN等)

※接続できない場合は、「ERROR」がDBのステータスに記録されます。



今回のTIPからは、以下の事を確認しました。

- ✓ DB接続成否の監視方法
- ✓ DB接続成否の履歴確認



MaxGauge TIPsはこちら

[http://www.ex-em.co.jp/exem\\_lab0/exemplabo\\_maxgauge\\_tips\\_index.html](http://www.ex-em.co.jp/exem_lab0/exemplabo_maxgauge_tips_index.html)

<お問い合わせ>

日本エクセム株式会社

TEL : 03-4530-9598

e-mail : [service@ex-em.co.jp](mailto:service@ex-em.co.jp)

DATABASE **ARTIST** GROUP

<http://www.ex-em.co.jp>