How to modify your subscription shell for GISC Tokyo's system migration

GISC Tokyo plans to replace WIS portal system on 1st March 2018. At the same time, new system will start providing all the services via "https" instead of "http". The new system will take over the URL (www.wis-jma.go.jp/), and accepts both https and http access, but some users, who use the Downloading Script (generated via GISC Tokyo portal), need to change protocol and to add a "wget" command option in your script. The modified Downloading Script works for both current and new systems. We recommend you to change the script in advance, but by the end of February, 2018.

If you have any questions, please mail to wis-jma@met.kishou.go.jp.

- Steps are following:
  - Stop your script
    Example:
      Linux : $ sh ./download.sh stop
      Windows: PowerShell .\download.ps1 stop
  - Edit your script
    Details are shown in "Point for Modify" below (Changes were highlighted in Yellow)
  - Restart your script
    Example:
      Linux : $ sh ./download.sh start > /dev/null 2>&1 < /dev/null &
      Windows: PowerShell –WindowStyle hidden .\download.ps1 start
  - Check

Please look into your local directory for downloaded files. New files will be stored there.

- Resources

  Introduction to DAR at GISC Tokyo (Tutorial Slides)

  http://www.wis-jma.go.jp/cms/file/Introduction_to_DAR_GISC_Tokyo_Rev04.pdf

- Point for modify

  ➢ Protocol of atomURL (from "http" to "https", one place)

  ➢ wget command option (add "--no-check-certificate", two places)

```sh
#!/bin/sh
### Configuration ###
atomURL="https://www.wis-jma.go.jp/data/syn?ContentType=Text&Category=Analysis&Type=Alphanumeric"
userName=""
userPassword=""
limitSpeed="128k"
sleepTime="30"
export PATH=$PATH:"/usr/bin"
workParentDirectory=`pwd`
storeDirectory=$workParentDirectory"/from_WIS-JMA"
scriptName=${0##*/}
workDirectory="_"$scriptName
formerETagFile="formerETag"
pidFile=$workDirectory"/pid"
######
loopSubscribe(){
```

```
while [ 1 ]
do
    nowTime=`date -u +'%H%M%S'`
    tmpFilePrefix="_"$nowTime"_"
    tmpFiles="_"
    rm -f $tmpFiles*
    ### download list of published data ###
    updatedListFile=$tmpFilePrefix"UpdatedList.txt"
    updatedListLogFile=$tmpFilePrefix"UpdatedListLog.txt"
    if test -s $formerETagFile
    then
        formerETag=`cat $formerETagFile`
        header="If-None-Match: "$formerETag
        stdout=`wget --no-check-certificate --header="$header" --header="Cache-Control: no-store" --save-headers -T 60 -t 1
-nc --limit-rate="$limitSpeed" -o "$updatedListLogFile" -O "$updatedListFile" "$atomURL" 2>&1`
    else
        stdout=`wget --no-check-certificate    --header="Cache-Control:  no-store"  --save-headers  -T  60  -t  1  -nc
--limit-rate="$limitSpeed" -o "$updatedListLogFile" -O "$updatedListFile" "$atomURL" 2>&1`
    fi
    ### download published data ###
    if test -s $updatedListFile
    then
        filesLogFile=$tmpFilePrefix"FilesLog.txt"
```

```bash
        stdout=`wget --http-user="$userName" --http-password="$userPassword" --no-check-certificate -T 60 -t 10 -nc
--limit-rate="$limitSpeed" -o "$filesLogFile" -i "$updatedListFile" -P "$storeDirectory" 2>&1`

        eTagValue=`grep "ETag:" $updatedListFile | tail -1 | cut -d" " -f2`
        if test -n "$eTagValue"
        then
            echo $eTagValue > $formerETagFile
        fi
    fi

    sleep $sleepTime

    done
}

startSubscribe(){
    if test ! -d "$workDirectory"
    then
        mkdir -p "$workDirectory"
    fi
    if test ! -d "$storeDirectory"
    then
        mkdir -p "$storeDirectory"
```

```
    fi
    if test -s $pidFile
    then
        pidFileValue=`cat $pidFile`
        if kill -s 0 $pidFileValue 2>_pErr
        then
            echo "script["$scriptName"] is already running. Please stop or restart script["$scriptName"]."
            exit 0
        else
            rm -f $pidFile
            echo $$ > $pidFile
            cd "$workDirectory"
            loopSubscribe
        fi
    else
        echo $$ > $pidFile
        cd "$workDirectory"
        loopSubscribe
    fi
}

stopSubscribe(){
    if test -s $pidFile
```

```
    then
        pid=`cat $pidFile`
        kill -9 $pid
        rm -f $pidFile
    fi
}

case "$1" in
    start)
        startSubscribe
        ;;
    stop)
        stopSubscribe
        ;;
    restart)
        stopSubscribe
        startSubscribe
        ;;
    *)
        echo "Usage: sh ./"$0' {start|stop|restart}"
esac
```