GSM (Unix) and cron

1. Introduction

This document describes the use of GSM (Unix) in conjunction with the Unix **cron** facility. A description of cron is beyond the scope of this document.

IN ORDER TO RUN GSM (UNIX) SUCCESSFULLY FROM A cron SCRIPT THE BACNAT VARIANT MUST BE V3.340, OR LATER.

2. Assumptions

It is assumed that cron (and crontab) have been configured to run a specific script at regular intervals. For the purposes of this document an example script **glexport,** which can reside in any directory (i.e. it does not have to reside in the $GLDIR directory) will be used to illustrate the use of GSM (Unix) with cron:

```
echo "date at start of cron job: \c" >> /tmp/gsmcronlog
date >> /tmp/gsmcronlog

GLDIR=/global/test/global
export GLDIR
PATH=$PATH:$GLDIR/bin
export PATH
GLTERM=crontap
export GLTERM
GLNAME=pyex
export GLNAME

echo "Starting GSM from cron..." > /tmp/gsmlog
date >> /tmp/gsmlog
global -r -j -d >> /tmp/gsmlog

echo "date at end   of cron job: \c" >> /tmp/gsmcronlog
date >>/tmp/gsmcronlog
```

3. The glexport Script

This section describes the various features of the example script.

3.1 Logging

The lines:

```
echo "date at start of cron job: \c" >> /tmp/gsmcronlog
date >> /tmp/gsmcronlog
echo "date at end   of cron job: \c" >> /tmp/gsmcronlog
date >>/tmp/gsmcronlog
```

are not strictly necessary but merely provide a convenient way to log, in the file /tmp/gsmcronlog, a record of each time the script is run. Note this file is always appended to, and contains a complete history of all invocations of the example script. This log file is expected to grow in size very slowly but nevertheless it should be created on a filing system with plenty of free space.

Similarly, the lines:

```
echo "Starting GSM from cron..." > /tmp/gsmlog
date >> /tmp/gsmlog
```

are not strictly necessary but log the start of each GSM session in the file /tmp/gsmlog. Note that the "gsmlog" file is recreated every time the script is run. This log file may grow in size very quickly (depending on the screen output of the various GSM tasks). Thus, it should be created on a filing system with plenty of free space.

3.2 Establishing the Environment

The lines:

```
GLDIR=/global/test/global
export GLDIR
PATH=$PATH:$GLDIR/bin
export PATH
GLTERM=crontap
export GLTERM
GLNAME=pyex
export GLNAME
```

establish the various shell variables required by GSM:

GLDIR Must be set to directory of an installed GSM (Unix)

PATH Must include the "bin" directory of an installed GSM (Unix)

GLTERM The Unix terminal type specified by this environment variable must be mapped to a valid GSM TAP in the Systems file (see section 4). Note that a command line parameter is now available to supply the Unix terminal type (see section 6). This setting is not strictly required as the TERM environment variable will used as a last resort. However, on some versions of Unix the TERM variable may be unpredictable when running via cron;

GLNAME The Unix user name specified by this environment variable must be mapped to both a valid System-id and GSM user name in the Systems file (see section 4). Note that a command line parameter is now available to supply the Unix user name (see section 6). This setting is not strictly required as the Unix user name will used as a last resort. However, on some versions of Unix the user name may be unpredictable when running via cron

3.3 Running GSM (Unix)

The line:

```
global -r -j -d >> /tmp/gsmlog
```

runs the GSM (Unix) user. The GSM (Unix) BACNAT version must be V3.335, or later. **HOWEVER, YOU ARE STRONGLY ADVISED TO USE V3.340, OR LATER**.

In this example, the following command line parameters are used:

-j	This option is **mandatory** when GSM (Unix) is run from a cron script;
-r	To avoid any possible problems with Unix permissions you are strongly recommended to include this option to allow global to run as user *root*;
-d	This option enables diagnostics which may be useful if problem trouble-shooting is required.
>> /tmp/gsmlog	You are advised to redirect all diagnostic messages and GSM output to a log file. In this example, the messages are appended to the log file freshly created by the glexport script.

4. The Systems file

The GSM (Unix) Systems file must contain the appropriate options to allow the "cron user" to run GSM (Unix).

4.1 Mapping the Unix User to a GSM System

The following line is required to map the Unix user (in this example pyex), specified by the GLNAME setting in the glexport script, to a valid GSM System (in this example 0x20):

```
SYSTEM  0x20 USER 1 LOGNAME pyex
```

4.2 Mapping the Unix User to a GSM User

The following line, in the OPIDMAP section of the Systems file, is required to map the Unix user (in this example pyex), specified by the GLNAME setting in the glexport script, to a valid GSM operator-id (in this example PYEX):

```
pyex    PYEX
```

The Unix user **MUST** be mapped to a GSM user via the Systems file or else the Auto-Run GSM user will wait at the OPERATOR-ID prompt.

4.3 Mapping the Unix Terminal Type to a GSM Terminal Code (TAP)

The following line, in the TERM section of the Systems file, is required to map the Unix terminal type (in this example crontap), specified by the GLTERM setting in the glexport script, to a valid GSM Terminal Code (in this example 1 - the meaning of the slightly unusual $.1 is explained below):

```
crontap 1
```

The Unix terminal type **MUST** be mapped to a GSM Terminal Code via the Systems file or else the Auto-Run GSM user will wait at the TERMINAL TYPE prompt.

Because the output of the GSM session started by cron does not appear on a screen the GSM Terminal Code is not that important. For **some simple applications**, the basic teletype terminal code (i.e. 0) may be used but for most applications (and all Speedbase applications) a terminal code with support for cursor positioning will be required. In general any terminal code may be used providing the TAP file (e.g. $.167) is present on SYSRES and, for Speedbase applications, the Speedbase TAP file (e.g. T>167) is also present on SYSRES.

All real TAP's include screen-specific escape sequences, of one form or another, that may complicate the casual inspection of the log file created by the GSM session. In order to create a log file that just contains displayable ASCII characters a special TAP $.1 has been created. This TAP file (and the equivalent Speedbase TAP file, T>1), which are of no use for a real screen or emulator, will be available with GSM SP-17.

5. Starting GSM Tasks

The combination of the environment variables established in the glexport script and the settings in the Systems file will allow the Auto-Run cron user started by the cron process to auto-login to GSM (Unix). Unless further action is taken, the Auto-Run user session will be waiting at the top level menu (or GSM READY: prompt if a Start Menu/Program has not be configured in $CUS).

Although the task(s) to be performed by the Auto-Run cron user could be started via type-ahead (i.e. using the -t*type_ahead_string* global command line option) you are strongly advised to use a Start menu entry. $AUTH, $PASSWD and $AUTH32 all allow a user to auto-run a particular menu entry. Don't forget to configure the appropriate Authorisation Program (i.e. $AUTH, $PASSWD or $AUTHEX) in $CUS as well as adding a special Auto-Run user (PYEX in this example).

You are advised to include all the GSM commands to be run by the Auto-Run user in a job (and specify the job as the target of the Auto-Run menu entry) rather than running a string of commands via type-ahead in the Auto-Run menu entry.

DON'T FORGET TO MAKE THE FINAL LINE OF THE JOB A COMMAND TO RUN $BYE TO TERMINATE THE AUTO-RUN GSM SESSION.

5.1 Multiple Partitions

You are strongly advised to run all the GSM tasks in partition 1 of the Auto-Run cron user. Although, it is possible to run several tasks in multiple partitions of the Auto-Run user, timing issues may introduce problems.

Also, if the task(s) complete quickly an artificial delay may have to be introduced to allow the background partitions (i.e. P2, P3 etc.) to initiate completely before running $BYE (in P1) to terminate the session.

6. New global Command line options

As described in section 3.2 the environment variables GLTERM and GLNAME are required to supply explicit Unix terminal and user name parameters. Two new global command line parameters are now available to achieve the same effect:

-o*unix_user_name*
-q*unix_terminal_code*

For example the following lines in the example script:

```
GLTERM=crontap
export GLTERM
GLNAME=pyex
export GLNAME
global -r -j -d >> /tmp/gsmlog
```

could be replaced by:

```
global -r -j -opyex -qcrontap -d >> /tmp/gsmlog
```

Important note: There is no space character between the -o or -q and the associated text string parameter.

7. GSM Logging

In addition to the simple display log file described in section 3.1 it may also be necessary to log events using GSM Event logging as described in the V8.1 GSM Utilities Manual. If Event Logging ($LOG) is enabled then the log file must be purged regularly to prevent the fixed-sized log-file from filling up.

A number of techniques in GSM (Windows) to support "spoofed users" provide various methods to monitor the progress of applications run by the "spoofed user". Some of these techniques may be ported to GSM (Unix) in future GSM Service packs.

8. Other Issues

The Auto-Run user initiated by a cron job counts as a single user as far as GSM User Count Licencing is concerned. You must ensure that the total number of "normal" users logged to GSM is always less than the allowed User Count when the cron job executes.

A number of techniques in GSM (Windows) to support "spoofed users" provide various methods to increase the total User Count to allow for "text-only" background users. Some of these techniques may be ported to GSM (Unix) in future GSM Service packs.