Tips for Setting up Cron Jobs for Fitrix programs

Cron jobs are very useful for tasks you need to perform when you do not want to do it yourself or you need it done at a specific time. Cron does not forget – it always runs your script, but the script may not run correctly. This document does not describe all the ins and outs of setting up cron. There is enough written about that already. This document does describe tips to get Fitrix programs to run via cron. Below is a summary of tips.

- Program must have no user interaction (selection criteria windows or prompts)
- Although not required, you may find it useful to setup a user id specifically for the cron jobs. If you have to track down an issue and you see it is caused by the cron user id, you can eliminate a lot of leg work.
- You have to setup every little bit in the environment. The cron default environment is very sparse. Leaving out a variable setting can cause your script to fail even if it runs fine on the command line even when you are logged in as the cron user. Centralize this environment setup and do not forget FGLGUI=0.
- Setup your script in cron using crontab -e

Program must have no user interaction

If the program you want to run via cron has any user interaction, the program will just hang and you'll never see any output. If there are selection criteria windows that are called from ml_filter, you can bypass the ml_filter function by adding the filter '<sql filter>' on the command line like so:

fglrun o_custls.42r -d sample filter '1=1' order 'strcustr.cust_code'

The command runs the customer summary by code for all customers when run in the $fg/accounting/ar.4gm/o_custls.4gs directory.

If the program has windows and prompts throughout or you need to run the ml_filter function , then you need to do some coding. Typically, you will add a command line parameter such as –auto or –cron. You check for this parameter before you call the window or prompt. Below is snippet of such code (from $fg/accounting/ar.4gm/p_settle.4gs/midlevel.4gl):

```
        let p_error = false
        let p_errorText = null
        let p_errorText_d = null
        let prompt_response = "Y"

        if not get_arg("-cron")
        then
            let sel_filter = sel_filter clipped, " and ",  sel_inv()
        end if

        #_ccard_flow
        call ccard_flow()
```

```
        # FGLPP END   after block ml_filter sel_filter

end function
# ml_filter()
```

Setup a user id for cron jobs

It is useful to setup a separate user id for cron jobs. If you have to track down an issue and you see it is caused by the cron user id, you can eliminate a lot of leg work. You can also centralize all the application related cron scripts in the cron user's bin.

Setup everything in the environment

The cron environment is very sparse in that it is not the same as when you log in as a user. You must setup all environment variables. If you centralize this environment setup to a single script you can source (dot) it from all your cron scripts. This makes adding cron scripts easier and much more maintainable.

Below is a bare bones environment setup for a fitrix production environment. There are a couple of things make note of. You must use ***full path names*** to all files. You must set FGLGUI=0 so that fitrix reports and posting programs can run via cron. You want to add all site specific variables in this centralized script so that you do not have to change multiple scripts every time you change something.

```
#!/bin/sh

# Sets the default environment variables to production
. /fitrix/bin/env_prod.sh

# live database
DBNAME=live
export DBNAME

# use default system file creation mask (umask)
umask 000

# Set TERM and TERMCAP, May vary depending on your installation
export TERM=ansi
export TERMCAP=/var/opt/fitrix/ifmx_ids11/etc/termcap

# For reports that run in gui mode
# Setting FLGGUI=0 forces no client interaction so these can
# be run via cron
export FGLGUI=0
```

If you called this script cronenv.sh and you had it in the bin of the cron user id of fxcron, you source it into your cron script like so:

```
#!/bin/sh
# Sets the default environment variables to production
```

```
. /home/fxcron/bin/cronenv.sh

# date extension for output
fdate=`date +%Y%m%d`

# use run_prg.sh to run program
run_prg.sh $fg/accounting/ar.4gm/o_custls.4gs -d $DBNAME \
    filter '1=1' order 'strcustr.cust_code' \
    destin /home/fxcron/cronout/o_custls.${fdate}
```

Notice the use of run_prg.sh to run the program. This is a script distributed in /fitrix/bin. Also notice the redirection of output and the output file name with the date extension. The default output of fitrix programs is a file named 'report.out' in the program directory. If you ran a posting program every night at 10:30pm you would overwrite report.out every night. You change this with the destin parameter followed by the output file name.

Setup cron to run your script

Log in as the cron user and use the '`crontab –e`' command to add your script to the cron jobs for the cron user. When you run the crontab –e (for edit) command for the first time are put into the default editor (usually vi) and the file may be empty. You may find it useful to add the cheat sheet below to the top of the file to remind you what the first 5 parameters are. The 6th parameter must be the ***full path name*** to your script.

```
#              field          allowed values
#              -----          --------------
#              minute         0-59
#              hour           0-23
#              day of month   1-31
#              month          1-12 (or names)
#              day of week    0-7 (0 or 7 is Sun, or use names)
#
00 04 * * * /home/fxcron/bin/o_custls.sh  > /tmp/o_custls.sh.cron 2>&1
```

The asterisks above mean any or all days of the month, every month and all days of week. The o_custls.sh script is set to run at 4:00am every day. As for the '> /tmp/o_custls.sh.cron 2>&1' part. This redirects any screen output to the /tmp/o_custls.sh.cron file. The 2>&1 redirects standard error to standard out so errors are also redirected to /tmp/o_custls.sh.cron. If you do not get your intended results, you can check the file to which you redirected standard out and standard error for clues.