# *Setting Up Required Resources*

In this chapter, we explain how to set up your ProMAX® software license and the ProMAX® Development Environment.

**In This Chapter**

# Installing the Software License (FLEXlm)

FLEXlm is a popular licensing software package and is used by many vendors, including Landmark. Landmark has enhanced the licensing software to include more features; this superset is called LAM.

Because of all the available features in FLEXlm, we can more accurately license our software to meet your individual requirements.

You need a license for the ProMAX® software release. See ***Obtaining a License*** below.

If you have other software using FLEXlm, refer to *Installing FLEXlm for Current FLEXlm Users* in this section. Otherwise, create a license.dat file.

## A Look Ahead

This section includes the following:

*Obtaining a License*- a list of what you need to request your FLEXlm license.

*Before Installing FLEXlm*- a list and explanation of what to include in your license.dat file.

*Installing FLEXlm*- step-by-step instructions for installing and starting FLEXlm.

*Installing FLEXlm for Current FLEXlm Users*- suggestions for running more than one licensed product with FLEXlm.

*Starting FLEXlm*- ways to make it easier to start FLEXlm.

*Command Summary*- explanation and syntax of some of the commands provided with FLEXlm.

## Obtaining a License

Before you install FLEXlm, obtain a license by contacting Landmark in one of the following ways:

- e-mail us at license@lgc.com
- the World Wide Web (http://www.lgc.com)

- telephone the Houston Licensing Group at 281-560-1110
- telephone the London support center at + 44 (0) 1932 832-100

You must provide your lmhostid, which you can obtain by running the following lmhostid command from the computer where the ProMAX® software is loaded:

**$PROMAX_HOME/sys/bin/flexlm/lmhostid**

The output will look similar to the following:

lmhostid - Copyright (c) 1989-2006 Macrovision Europe Ltd. and/or

Macrovision Corporation. All Rights Reserved.

The FLEXnet host ID of this machine is "XXXXXXXXXXXX"

If you do not have your FLEXlm software yet, you can still obtain a license. First, you need to type one of the following commands at the UNIX prompt:

- For RS/6000 systems, type

  **uname -m**

- For Solaris systems, type

  **hostid**

- For SGI systems, type

  **/etc/sysinfo -s**

- For Linux systems, type

  **hostid**

Each of these commands will reply with a number; call us with this number and ask us for a license. The telephone number inside the America's is 281-560-1110 (option 6). Outside the America's, the number is 44-1-932-83-2100. If you e-mail the information, send it to license@lgc.com. Be sure to type **License Generation** on the e-mail subject line. Additionally, within the mail message, include the command you used to generate this number.

## Before Installing FLEXlm

You need to obtain a license.dat file containing a minimum of three lines with the following syntax:

**SERVER [hostname][ lmhostid] [port]**

**DAEMON [program_name][ path_to executable]**

**FEATURE [product][ daemon][ version][ expdate][ number][encryption] [ “productco”][ hostid]**

The following is an example that follows the stated syntax:

**SERVER sun2000 80710ac2 7394**

**DAEMON licsrv /advance/sys/bin/flexlm/licsrv**

**FEATURE PROMAX2D licsrv 6.000 1-jan-00 0 DBAEC0815BD243FF73F0 “Landmark” 80710ac2**

Starting with ProMAX® Release 1998.0, your license will have included an *increment* stanza instead of a *feature* stanza. That syntax looks as follows:

**INCREMENT PROMAX2D licsrv 1998.0 01-jan-00 0\ DBAEC0815BD243FF73F0 “Landmark” 80710ac2**

### License File Tips

The following tips are provided regarding your license file:

- Licenses require a sales order, which was issued by your sales representative.
- Unless we receive other information, SERVER lmhostids are tied to the CPU running the ProMAX® software.
- The DAEMON licsrv stanza must contain the actual path to $PROMAX_HOME/sys/bin/flexlm/licsrv or other Landmark licsrv.
- Do not wrap the FEATURE or INCREMENT stanza onto a second line.

### **SERVER hostname lmhostid port** (Line 1)

The SERVER line specifies the FLEXlm license host, a system specific CPU host ID number, and a port number.

The hostname is the hostname of the system running the FLEXlm license server daemon. The lmhostid is the value returned by the lmhostid command. Refer to *Obtaining a License* in this section for details.

FLEXlm uses a port number to communicate. If you are already using FLEXlm, you can continue using your current port number. We pick a number that is typically not used. If you get an error stating that your port number is in error, look in /etc/services and choose an unused number greater than 1023. But do not add an entry into /etc/services.

***Note:*** The hostname and the lmhostid must match the system that the license server will be running on. We assume that these are from the same system and that this system is the one that will be running the ProMAX® software. If they are not the same system, let us know before we generate your license file.

**DAEMON program_name path_to_executable** (Line 2)

The daemon line specifies the program name (daemon) that will manage the license for the product; all Landmark products use the program name licsrv. The daemon line also specifies the path to the daemon executable.

**FEATURE prod daemon version expdt encryp “prodco” hostid** (Line 3)

The Feature or Increment line lists the products you purchased; you need one feature line per product.

**product**: The ProMAX® product being licensed such as PROMAX2D.

**daemon**: The name of the license server daemon; Landmark uses licsrv.

**version**: The version of the licensed product.

**expdate**: The date is used when issuing temporary licenses. After this date the license will no longer be valid. The date 1-jan-0000 indicates a permanent license which will not expire.

**encryption**: This is based on all the information listed on the feature and server lines.

**"Product Company"**: This will typically be **Landmark** to indicate this feature is from Landmark Graphics.

**hostid**: This the lmhostid of the system (for example, **80710ac2**) that is licensed to run this FEATURE.

## Installing FLEXlm

If you are currently running FLEXlm for other products, you can skip this section and proceed to the *Installing FLEXlm for Current FLEXlm Users* section. Otherwise, you can install one of three types of licenses: permanent, temporary, or demo. Temporary licenses are used to evaluate products before purchasing, and they expire after a given time period. Permanent and temporary licenses are treated exactly the same for purposes of installation. Demo licenses also expire after a short period of time. Demo licenses are typically used for sales demonstrations and are not generally given to working systems.

It is recommended that you shut down any previous ProMAX® license server and start the license server delivered with the OpenWorks. It will be necessary to edit the DAEMON line in your new license.dat file to match the location where LAM was installed.

### Installing Permanent and Temporary Licenses

1. Create the license file from the information we gave you. The license manager and your software product must know where the license file is located. FLEXlm searches for the license file in the following order:

- First, it will look in $OWHOME/lam/license.dat.

  OWHOME is the location of the OpenWorks software or the target directory of the software you are loading. To use this you must have the environment variable OWHOME defined. If OWHOME is not defined, The ProMAX® system defines it to be $PROMAX_HOME/port/OpenWorks.

- Next, it will look for the environment variable LM_LICENSE_FILE, as defined in your start up script, .profile or .login file. $OWHOME/lam/license.dat is appended to any search path that has been previously set.

- Finally, it will look in the /etc/license.dat file.

2. Start the license server:

```
% /lam_directory/lam/bin/lmgrd -c LICENSE_FILE > LOGFILE&
```

   /lam_directory is the location of where the lam product was installed from the OpenWorks CD. LICENSE_FILE is the path to the license.dat file and you write FLEXlm messages to LOGFILE. FLEXlm also writes a file /usr/tmp/locklicsrv. Therefore, the directory /usr/tmp must must exist and you must have write permissions to start FLEXlm.

   If you experience difficulties starting FLEXlm, first look on the LOGFILE for messages. You may also set the environment variable err_level to 0:

```
setenv ERR_LEVEL 0
```

   This command will generate verbose error messages, which should help to identify the location of the problem.

### Installing Demo Licenses

Demo licenses do not need a license server. Simply place the license in /etc/license.dat or $OWHOME/lam/license.dat, or point the environment variable LM_LICENSE_FILE to the location of the file.

## Installing FLEXlm for Current FLEXlm Users

Many software packages use FLEXlm. You can add the ProMAX® license file to your existing license file with the following steps:

1. Open your existing license file.
2. Check to see that the lmhostid on the SERVER line is identical to the SERVER line in your existing file. If it is not, the encryption on the feature line will not be correct. Send us a copy of your existing server line(s), and we will provide you with properly encrypted features.
3. Place the DAEMON line from your ProMAX® license file with the existing DAEMON lines. Remember, if you have multiple vendors providing products, you will have multiple DAEMON lines. However, if your other FLEXlm products are from Landmark and use the licsrv daemon, you will not need another DAEMON line.
4. Add the INCREMENT line from your ProMAX® license file.
5. Signal the license program to re-read the license file:

```
/lam_directory/lam/bin/lmreread -c LICENSE_FILE
```

## Starting FLEXlm

You can create aliases or write scripts to make it easier to start FLEXlm, keeping in mind that only one daemon should be running on the server system. You should consult with your system administrator before attempting to have FLEXlm start automatically during the boot process.

### .flexlmrc file

The first time you use FLEXlm, it creates a hidden file: .flexlmrc. This file contains a speed-optimization to help FLEXlm locate the license file and the license server. You need to remove file before changing your configuration.

Landmark suggests that you disable the creation of this file by including the following line in your environment file:

```
setenv FLEXLM_NO_CKOUT_INSTALL_LIC 1
```

You will still want to use the variable:

```
setenv LM_LICENSE_FILE /your/license.dat
```

### Aliases

You can include aliases in your .cshrc file to make it easier to start FLEXlm. For example, if you set the following alias and then type **start** at the command line, the **lmgrd** command will get executed.

```
alias start ‘ /lam_directory/lam/bin/lmgrd -c /etc/license.dat > /tmp/log &
```

### Scripts

You can write or edit a script to start FLEXlm automatically during the boot process, or you can write a script to execute from the command prompt after the system comes up. Remember to consult with your system administrator before attempting to have FLEXlm start automatically during the boot process. An error in the FLEXlm start up script could result in a failure to boot.

The following script is an example which can be used to automatically start FLEXlm when Linux boots:

```
#==========lmgrd script==========

#!/bin/ksh

#

# description: lmgrd - This script will start and stop the Landmark license
manager

#

# chkconfig: 345 85 85

#

# Source the library functions

. /etc/rc.d/init.d/functions

PROG="<path to $PROMAX_HOME>/sys/bin/flexlm"

BASE="<path to directory where license.dat resides, like
$PROMAX_HOME/etc"

# let see how we were called

case "$1" in
```

```
start)
    echo "Starting Landmark license manager: "
    if [ -x $PROG/lmgrd ] ; then
        daemon $PROG/lmgrd -c $BASE/license.dat -l $BASE/license.log
        echo
    fi
;;
stop)
    echo "Shutting down Landmark license manager: "
    if [ -x $PROG/lmdown ] ; then
        daemon $PROG/lmdown -c $BASE/license.dat -all
        echo
    fi
;;
restart)
    echo "Restarting Landmark license manager"
    $0 stop
    $0 start
    echo "done."
;;
*)
    echo "Usage: lmgrd {start|stop|restart}"
    exit 1
esac
```

To setup your system to automatically start the license manager, perform the following steps as root:

1. Create lmgrd script file and place it into the /etc/init.d directory.
2. Change permissions on this file as follows:

   **chmod 755 lmgrd**

3. Edit the lmgrd file to update the locations for $PROMAX_HOME and location of license.dat:

   **PROG="path to $PROMAX_HOME>/sys/bin/flexlm"**

   **BASE="path to directory where license.dat resides" (such as $PROMAX_HOME/etc)**

4. Set up the lmgrd script to start at boot:

   **/sbin/chkconfig lmgrd on**

5. Try to restart lmgrd using the lmgrd script:

   **/sbin/service lmgrd restart**

Restarting is an excellent way to verify the integrity of the scripts before rebooting your system. If there are any errors in the scripts, it is best to correct them now.

## Command Summary

These are a few other utilities of note supplied with the FLEXlm package. The brackets designate an optional setting.

- **lmdown** *[-c LICENSE_FILE]*

  This utility will bring down the license server.

- **lmgrd** *[-c LICENSE_FILE]*

  This utility is the license server; it grants licenses as described by license.dat file.

- **lmhostid**

  This utility will return the license manager host id for the local machine.

- **lmremove** *[-c LICENSE_FILE] feature user> host [display]*

  This utility allows the system administrator to remove a single user's license for a specified feature. This utility is useful if, for some reason, a license becomes unusable. This command will free the license and return it to the license pool.

Where

*[LICENSE_FILE]:* The absolute path (filename inclusive) to the license.dat file.

*feature:* The feature to free, such as PROMAXUI, PROMAX3D.

*user*: The username currently possessing the license.

*host*: The host that you are on.

*[display]*: If you have more than one license of this feature on a specified host, use display to differentiate between them.

- **lmreread** *[-c LICENSE_FILE]*

  This utility uses the license.dat file to be reread by the lmgrd server. lmgrd allows changes in the license.dat file to be propagated without bringing the server down.

- **lmstat** *[-a] [-A] [-c LICENSE_FILE] [-S daemon] [-f feature] [-l regular_expression] [-s server] [-t value]*

  This utility gives you statistics pertaining to the license server.

  Where

  *[-a]*: Display everything.

  *[-A]*: List all active licenses.

  *[-c]*: Specify a license file.

  *[-S]*: List all users of <daemon>'s features.

  *[-f]*: List all users of <feature>.

  *[-l]*: List all users of matching license(s).

  *[-s]*: Display status of server node(s).

  *[-t]*: Set lmstat timeout to value.

- **lmver** *[flexlm_binary]*

  Where the flexlm_binary is **licsrv**

  This utility reports the current version of FLEXlm.

# Configuring ProMAX® PD

This section discusses PD configuration and debugging.

## PD Configuration

PD (Pointing Dispatcher) is used by MVA tools such as the interaction of the MVA XSec Window, and the Interactive Horizon RMO Analysis, or View CRP gather process. It is not used by the Flow Chart. PD is also used by other ProMAX® tools such as IDA (Interactive Data Access), in Disk Data Input, along with Trace Display, FK Analysis, Interactive Spectral Analysis, Interactive Radon Analysis, Velocity Analysis, Volume Viewer/Editor, and Make Database Basemap. The following are recommendations in using PD: (See also *PD in an Integrated environment for Release 2003.X* if you work in an integrated environment).

### Starting PD

If PD is not running on your workstation, any process needing PD will automatically start PD. If you find that PD is not starting automatically, it can be due to potential licensing problems or timing issues between when PD starts and when interactive processes start. It is then recommended to start PD manually before running any ProMAX® process. PD can be started via the command line. Set the PROMAX_HOME environmental variable, then use the following command:

```
% $PROMAX_HOME/port/bin/start_pd
```

Execute this command every time the machine is rebooted. Please refer to *Installing the Software License (FLEXlm)* in the *Setting Up Required Resources* chapter of the ProMAX® *System Administration Guide* for an example of starting a process at boot time.

### Running with NIS

NIS is the Network Information Services, which used to be called Yellow Pages. Most non-trivial networks use this. If the machine is running NIS, the system manager should go to the yp master and add the following line to the services file:

```
lgc_pd          2011/tcp        pd
```

The number does not have to be 2011, but it needs to be unique and larger than 2010. If there is already an lgc_pd entry in the services file from other Landmark products, you do not need to add a second lgc_pd line.

After making the changes, the system manager needs to push the maps out on the network.

### Running without NIS

If you are running without NIS, add the following line to the /etc/services file for each machine needing PD:

```
lgc_pd          2011/tcp          pd
```

Again, make sure the numbers are unique. You will need to be root to edit the /etc/services file.

## Debugging PD Problems

If after installation PD fails to start when you are executing a process which uses PD, you should have an error message like this in the job.output:

```
Could not connect to PD server.
PD may not be running. Starting PD.
-rwxrwxr-x    1 promax   user       56280 Apr 17 1995
/advance/sys/bin/promaxpath
Setting OWBIN to /advance/sys/exe/lgc
The file /tmp/.pd.lck file exists which indicates that PD may already be
running. However, PD will be started anyway. If PD was already running, this
may cause problems.
xpd.c:310 300000000 xpdOpen: PDOPEN failed.
Could not connect to PD server even after attempting to start it!
```

If you have this error, type the following:

```
% ps -ef | grep pd
```

If no other PDs are running, you will get the following response to your command:

```
user 17988 15083   0 08:31:53  pts/0  0:00 grep pd
```

If PD is already running, you will get the following response to your command:

```
user 18016     1   0 08:33:02  pts/2  0:00 /advance/sys/exe/lgc/pd
user 18789 15083   1 08:33:18  pts/0  0:00 grep pd
user 20066 18016   0 08:33:02  pts/2  0:01 /advance/sys/exe/lgc/pd
```

Since the program cannot see these PD daemons, the program thinks PD is not running and tries to start it. If this happens, you will want to kill the two running PDs via the command line using stop_pd.

***Note***: Starting PD causes two to start; they operate in pairs, so stop the two PDs using the following command:

```
% $PROMAX_HOME/port/bin/stop_pd
```

Using the **stop_pd** command versus the unix **kill** command will also remove the **.pd.lck** file in the /tmp directory, otherwise you would have to remove that file manually. Then try rerunning the program and the PDs should start and operate normally.

If this was not the problem, type one of the following commands:

- If the system is running NIS, type:

  ```
  % ypcat services |& grep lgc_pd
  ```

  A non-trivial network will run NIS.

- If the system is mainly stand-alone or on a simple network, type:

  ```
  % grep lgc_pd /etc/services
  ```

The output should be:

```
lgc_pd        2011/tcp        pd
```

Make sure the number being used (2011 or whatever) is unique by typing one of the following commands:

- If the system is running NIS, type

  ```
  % ypcat services |& grep 2011 # Non-trivial network will run NIS
  ```

- If the system is mainly stand-alone or on a simple network, type

  ```
  % grep 2011 /etc/services
  ```

  The number does not have to be 2011, but it does have to be unique and larger than 2010.

If the problem still occurs, type the following two commands:

```
% cd $PROMAX_HOME/port/bin
```

```
% start_pd -l 0 -e PD.errors   # note "-l 0" is a zero, not the letter O
```

This causes the error level to be set to zero, which will produce verbose output. It also causes error messages to be written to the file PD.errors.

When this completes, wait 5 seconds for PD to write its error messages. Then send us the PD.errors file.

## PD in an Integrated environment

In an integrated environment (customers running both the OpenWorks and ProMAX® systems), The ProMAX® processing system is designed to use the new pointing dispatcher from OpenWorks. This is known as a ***Dynamic PD***. That is, one PD servicing all users. It requires the ***NetD*** running on the host where the Dynamic PD will be started and stopped. This host can be specified via the LGC_PD environmental variable. NetD is only distributed with OpenWorks, ***not*** with the ProMAX® software.

***Note***: For more information on OpenWorks PD, refer to Chapter 6: *Customizing Pointing Dispatcher* in the *Learning OpenWorks System Administration* guide.

For ProMAX®-only clients (not running OpenWorks) the ProMAX® software falls back to running it's own ***Static PD.*** That is, a separate PD service for each user.

To see what PDs you have running on your system, type the UNIX command:

```
% ps -ef |grep pd
```

If PD is running, your response can look like:

```
prouser 20335 20315  0 11:59:54 pts/5    0:00 grep pd

owuser 20199 20198  0 10:58:46 ?        0:00 pd
(OPENWORKS_500_owy2k@192.156.212.173:0_PD using 28000) -sname
28000 -autoexi

prouser 12724     1  0   Feb 08 ?        0:00 /apps/71/ProMAX/sys/exe/lgc/pd

prouser 12725 12724  0   Feb 08 ?        0:00 /apps/71/ProMAX/sys/exe/lgc/pd
```

This example shows a ProMAX® and an OpenWorks PD running by different users.

### Multiple Users on One System

In situations where several users are running concurrently on one system, you should set up each user account with its own PD server. This will prevent users from accidently exchanging data with each other through PD.

Use the following steps to create separate, unique PDs for each user:

1. For each ProMAX® user, edit the /etc/services or yp master file and add additional pd lines specifying different pd numbers. For example:

```
lgc_pd 3006/tcp        pd

lgc_pd13007/tcp        pd1

lgc_pd23008/tcp        pd2
```

2. In the user's profiles or start up script, set the LGC_PD_SERVICE environment variable to a unique pd service.

```
setenv LGC_PD_SERVICE lgc_pd1
```

3. In the user's profiles or start up script, set the LGC_PD environment variable to the service specified in the user's LGC_PD_SERVICE variable. The user's clients will connect to that service.

```
setenv LGC_PD :lgc_pd1
```

4. Start PD manually or via your user's script plus running a ProMAX® process that uses PD (like Volume Viewer/Editor). It will start up on the service specified by the variable LGC_PD_SERVICE. Check this by typing the UNIX command **ps -ef | grep pd**.

### Connecting Clients to a PD on Another System

If you are on a network system, you may have the case where one machine is a ProMAX®-only system (not running NetD but using Static PD and you need to read SeisWorks data from another machine running OpenWorks (running NetD and using Dynamic PD).

In order to connect to the Dynamic PD running on the OpenWorks machine, you must set the following

environmental variable on the machine running the ProMAX® software:

**setenv LGC_PD <hostname>:lgc_pd**

where <hostname> is the name of the machine in which you are pointing your $OWHOME. $OWHOME is set on the machine running only the ProMAX® software.

## Troubleshooting

With ***Static PD,*** the problem areas arise when multiple users attempt to access the same LGC_PD_SERVICE. These problems usually arise from not having environmental variables set properly. Also on busy networks/machines, or if the ProMAX® software crashes, the Static PD processes can become ***stale***.

# Configuring the ProMAX® Monitor (pmmon)

The ProMAX® monitor; Also known as pmmon is a program supplied in the sys/exe directory. pmmon is intended to monitor files being written by a ProMAX®/SeisSpace® flow and to suspend the flow to allow the user to take remedial action if any of the monitored files exceeds its configured size.

pmmon uses parameters from the config_file as follows :

**monitored file: filename Size in k, m or g bytes**

**monitored cycle time: Time in seconds**

**monitored filesize exit: Script searched for in port/bin**

* Filename : the filename relative to the flow directory to be monitored the pathname may include environment variable names indicated by a ˜$˜ prefix.

* Size : file size in bytes to trigger an exception may be followed by k, m or g to indicate kilo, mega or giga bytes.

* Time : frequency of monitor tests. The default is 30 seconds.

* Script : The program to call when an exception is triggered. That is, when a file size is exceeded. The default is to search for monitored_file_exit in port/bin.

When a file size is exceeded the flow is suspended by sending the flows process group the SIGSTOP signal and the script is called (in a separate process group). What happens when the script returns depends on the return code :

1. The flow is stopped by sending it a 'user terminate' signal.

2. The flow is resumed. This supposes that the file that triggered the exception has been shortened otherwise the exception will be retriggered immediately.

3. Or for any other return code the flow remains suspended.

The ProMAX® software includes a sample port/bin/monitored_file_exit script that attempts to send the user name from the running flow an email if it cannot decide that option 1 or 2 are intended. Note that it uses arguments

passed to tape_mount.exe to prompt for a response and captures that programs return code to indicate what is to be done based upon those arguments. While these return codes happen to be the same as the return options described above they are mot required to be so. It is the return code of the script that is important to pmmon. Please see the example script below.

If neither 1 nor 2 are returned to pmmon then the flow needs to be manually signaled to resume (or cancel) processing. The command to resume the flow is :-

**kill -s SIGCONT -- -{group id}**

Group id here is the process group leader of the processes that make up the flow. It is important to get this right otherwise the flow may not continue or may fail to clean up. To find the process group leader identify the bash shell running the appropriate shell script, the uxxxx or exec.x.qsh file, find {its process id} and then run :

**ps axo pgid,pid | fgrep {its process id}**

The other number in the return, or that process id if the numbers are the same, is the group id to use in the kill as long as the number is not 1.

Here are example config_file entries that result in the monitoring of the print out files from flows submitted by the ProMAX® or SeisSpace® user interfaces. In this example the files are limited to 1 megabyte each.

**monitored file: $JOBTMP/stderr 1m**

**monitored file: $JOBTMP/stdout 1m**

**monitored file: exec.$EXEVER.log 1m**

**monitored file: exec.$EXEVER.log.err 1m**

**monitored cycle time: 10**

**monitored filesize exit: monitored_file_exit**

To have pmmon called in flows built by the ProMAX®/SeisSpace® systems create a script port/bin/PromaxMonitor that contains :

#!/bin/bash
#
# return monitor setup for a ProMAX®/SeisSpace® batch request

```
#
echo -n ‘promaxpath sys/exe/pmmon‘
echo ’ $$ &>/tmp/jopmmon &’
```

Details of the construction of the flow script are available in the ProMAX® print page which explains the location of the PromaxMonitor call.

Finally here is the example port/bin/monitored_file_exit script :

```
#!/bin/bash
#
# RCS: ProMAX $Id: monitored_file_exit,v 1.4 2008/07/11
02:01:51 crs Exp $ $Revision: 1.4 $ $Date: 2008/07/11
02:01:51 $
#
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~
# monitored_file_exit - This script is called upon a monitored
file full condition
#       from the pmmon utility if it is run from the flow script.
#
# The standard function of this example script is:
#
# Popup the standard "tape_mount.exe" program using the
envar "DISPLAY"
# to direct the image to a working Xterm.  The error condition
is posted
# such that corrective measures can be taken and the job
continued or killed.
#
# Two buttons are presented: Button 1 (tape_mount.exe exit
value = 2) is
# set to unsuspend the task (see script below).  Button 2
(tape_mount.exe
# exit value 1) will send the TERM signal to the flow.
#
# If there is not an active Xserver running tape_mount.exe
will fail upon
# execution but this will cause job to remain suspended and
an email wil be sent
# to the user so corrective measures can be taken anyway.
#
# If tape_mount.exe does not pop a dialog,
```

```
# either the program is not found or that the envar "DISPLAY"
is not set.
# If the dialog pops-up but no text is visible, you may try
resetting your
# Xdefaults foreground and background.
#
# Standard usage: monitored_file_exit ARG1 ARG2
#
# ARG1 ($1) = "The message containging details of the file
identified as having
#             exceeded its monitored file size"
#
# ARG2 ($2) = process id of suspended ProMAX job.


#-------------------------------------------------------------------------
-
# tape_mount.exe will use $DISPLAY to determine which
terminal
# to show this message.  Reset DISPLAY envar.
#
#unset DISPLAY
#export DISPLAY=host:0.0

#-------------------------------------------------------------------------
# Set email recipient to someone other than the user
#
# Default is: $USER.  Set the envar
PROMAX_DISKFULL_MAILTO to someone other
#                   that the use if appropriate.
#
#                   Use "whoami" to make sure USER is set.
#                   Need to know which OS to run correct
"whoami".
#
# if PROMAX_HOME is undefined, determine from full
pathname of script
if [ -z "$PROMAX_HOME" ]; then
typeset dir=‘dirname $0‘
dir=${dir%*/port/bin}
export PROMAX_HOME=${dir}
fi
#
typeset EMAIL_TO
#
if [ -n "${PROMAX_MONITORED_MAILTO}" ]; then
```

```
EMAIL_TO=$PROMAX_MONITORED_MAILTO
else
EMAIL_TO=`whoami`
fi
#
#-------------------------------------------------------------------------
---
# Set-up display parameters so a tape_mount looks instead
like a disk message
#
# Command line args for tape_mount.exe:
#
#       1 = Dialog title message.
# 2 = Button 'one' message.
#       3 = Button 'two' message.
#       4 = ProMAX info.
#       5 = User info.
#       6 = Error message.
# 7 = comment string.
#
#                                       Dialog title message:
typeset DT_Msg="MONITORED FILE SIZE ERROR
CONDITION"
#                                       Button 'one' message:
typeset B1_Msg="CONTINUE FLOW - Only if you adjusted the
file!"
#                                       Button 'two' message:
typeset B2_Msg="KILL FLOW, Send flow a user termination
request"
#                                   ProMAX runtime information:
typeset RT_Info=" "
#                                                   User info:
typeset User_info=" User:$DFEUSER  On HOST:"`hostname`
#                                                  Error message:
typeset Err_Msg=" A ProMAX job PID:[$2] has exceeded a
monitored file size and is suspended."
#
#                               Execute tape_mount.exe pop-up
dialog.
typeset tapeMount=`promaxpath sys/exe/tape_mount.exe`
if [ -z "$tapeMount" ]; then
 tapeMount=$PROMAX_HOME/sys/exe/tape_mount.exe
fi
$tapeMount "${DT_Msg}" "${B1_Msg}" "${B2_Msg}"
"${RT_Info}" "${User_info}" "${Err_Msg}" "$1" DISK
&>/tmp/mtferr$$.log
```

```
#
typeset RES=$?
# Test return code: [2=button 'one' pressed, 1=button 'two'
pressed]
# but first is there an error ?
fgrep -q "ERROR" /tmp/mtferr$$.log
if [ $? -eq 0 ]; then
RES=99
fi
#
case $RES in
1) echo "Requesting Flow to exit"
   kill -s SIGCONT -- -$2
   kill -s SIGTERM -- -$2
   RES=1;;
2) echo "Unsuspending job: PID=$2"#
   kill -s SIGCONT -- -$2
   RES=2;;
*) echo "${Err_Msg}" "$1" | mail -s "Monitored File size
exceeded - ProMAX job suspended" ${EMAIL_TO}
   RES=0;;
esac
rm -f /tmp/mtferr$$.log
exit ${RES}
#         ---------- All Done ----------
```