PROP – Programmable Operator Facility

Andrew Bingham, 3/29/04
Information taken from the IBM v/VM CMS Planning and Administration Guide, V4R3.0, p.

Overview

The Programmable Operator Facility is a CMS program which runs on a virtual machine within z/VM. It intercepts all messages and requests routed to its virtual machine and handles them according to a set of preprogrammed actions. Primarily, this allows the workload on system operators to be reduced – message traffic can be reduced and routine tasks can be automated. Messages that cannot be handled by PROP are sent on to a designated human operator for action.

PROP is already installed on our system, and can be started from CMS in the following manner:

PROPST <rtable> [DISConn]

<rtable> is the name of the PROP RTABLE to be used.

DISConn causes PROP to be disconnected before it is started.

Example:
(as maint)

i cms
Ready; T=0.01/0.01 16:42:52
z/VM V4.3.0 2002-04-04 10:28
propst prop
Ready; T=0.01/0.01 16:42:57
PROP running - enter STOP to terminate

At this point, PROP is running, and all messages being sent to MAINT will be interpreted for action by PROP. Note that logging in as MAINT would not yield a working CP environment with PROP loaded. Typing STOP should terminate PROP (more on that *should* later)

The PROP Routing Table

PROP is setup using a PROP Routing Table, or RTABLE. This file contains the setup information required by PROP, as well as a list of messages to be listened for and actions to be taken in response to

those messages. For example, this is the default RTABLE included with z/VM:

```
00001 *        ----- SPECIFY THE PROP CONFIGURATION -----
00002
00003 * IDENTIFY THE LOGICAL OPERATOR
00004
00005 LGLOPR OPERATOR
00006
00007 * BLANK SEPARATOR IS '/', ARBCHAR SEPARATOR IS '$', "NOT" SYMBOL IS '¬'
00008
00009 TEXTSYM / $ ¬
00010
00011 * DO LOGGING WHEN THIS TABLE IS IN EFFECT
00012
00013 LOGGING ON
00014
00015 ROUTE    ----- END OF CONFIGURATION SPECIFICATION -----
00016
00017 *------------------------ --- --- -- -------- -------- -------- --------
00018 *T                       S   E   T  U        N        A        P
00019 *E                       C   C   Y  S        O        C        A
00020 *X                       O   O   P  E        D        T        R
00021 *T                       L   L   E  R        E        N        M
00022 *------------------------ --- --- -- -------- -------- -------- --------
00023 * FILTER OUT LOGON, LOGOFF, ETC. MESSAGES SO OPERATOR NEEDN'T SEE THEM
00024 *------------------------ --- --- -- -------- -------- -------- --------
00025 /OUTPUT OF                 19  27  3
00026 /LOGON                    19  23  3
00027 /LOGOFF$¬FORCED           19  80  3
00028 /DSCONNECT                19  27  3
00029 /RECONNECT                19  27  3
00030 /DIAL                     19  22  3
00031 /DROP                     19  22  3
00032 *------------------------ --- --- -- -------- -------- -------- --------
00033 * SEND FILTERED ASYNCHRONOUS CP MESSAGE STREAM TO LOGICAL OPERATOR
00034 *------------------------ --- --- -- -------- -------- -------- --------
00035                                   3                  DMSPOS   LGLOPR
00036 *------------------------ --- --- -- -------- -------- -------- --------
00037 * SEND A CP OR CMS COMMAND TO VM TO BE EXECUTED
00038 *------------------------ --- --- -- -------- -------- -------- --------
00039 /CMD /                     1   4    OPERATOR HOSTNODE DMSPOR   TOVM
```

Notice how the logical operator is set to OPERATOR; this means that if PROP encounters a message that doesn't match any of the defined text in the RTABLE, it will pass that message to OPERATOR.

Line 17 marks the beginning of the actual routing lines. These lines take the following format:

TEXT – The text string to be matched.

SCOL – The starting column in which PROP should look for the text string.

ECOL – The ending column in which PROP should look for the text

string.

TYPE – Identifies the origin of the message. Valid numbers are 1-8 and 30, with each representing a different message source.

USER - The userid of a user who is authorized to cause a certain action to take place. If blank, all uses match.

NODE - The network node id of a user who is authorized to cause a certain action to take place.

ACTN - The name of an action routine to be performed. Several defaults are included with CMS, and custom routines may be writing in assembly or as REXX execs.

PARM – A parameter, <=8 characters, that will be passed to the action routine.

It is important to remember that if a given message is not specifically handled in the RTABLE, it will be sent to the logical operators console.

Action Routines

Action routines are programs or execs that are executed when a message is matched with a routing table entry. IBM includes several default action routines with CMS, such as DMSPOR (miscellaneous functions), DMSPOS (route a message), and DMSPOL (load an RTABLE). We can also code custom action routines in IBM Assembly or REXX.

Log & Feedback Files

When LOGGING is set to 'ON' or 'ALL' in the RTABLE, PROP saves all messages to a log file. A feedback file is also kept, where users can insert feedback about system status, which can then be reviewed at a later time when a human operator is present..