OpenVMS Technical Journal

Effective Uses for the SYSGEN USER Parameters

Mark ‘Jilly’ Jilson

Proprietary Solutions Support Group

OpenVMS PCC Team

Overview

The OpenVMS SYSGEN utility includes parameters that a system manager can use to increase a system’s maintainability and increase application configurability. Integrating these parameters into system startup procedures may hasten resolution of system problems and can increase the flexibility of application startup.

SYSGEN USER Parameters

SYSGEN has four USER parameters that can be used in many ways, as determined by the system manager.

As shown in **Example 1**, there are two dynamic parameters, USERD1 and USERD2, and two non-dynamic parameters, USER3 and USER4. They all have the same defaults and limits and are numeric parameters. None of them can be used to store text values. These parameters are not referenced by any other part of OpenVMS. The parameters can be maintained like any other parameter using the MODPARAMS.DAT and the AUTOGEN command procedures. These parameters also can be set at the SYSBOOT> prompt during a conversational boot. You can determine the current active value of the parameter by examining the DCL lexical F$GETSYI, as shown in **Example 2**.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SHOW USER
Parameter Name      Current    Default     Min.      Max.     Unit  Dynamic
--------------      -------    -------    -------   -------   ----  -------
USER3                     0          0         0         -1
USER4                     0          0         0         -1
USERD1                    0          0         0         -1            D
USERD2                    0          0         0         -1            D
SYSGEN> HELP SYS_PARAMETERS USER
Sys_Parameters
  USERD1
       USERD1 is reserved for definition at the user's site. The
           reserved longword is referenced by the symbol SGN$GL_USERD1.

       On Alpha systems, this symbol is in the
       SYS$LOADABLE_IMAGES:SYS$BASE_IMAGE module.

       On VAX systems, the symbol is in the SYS$SYSTEM:SYS.STB module.

       USERD1 is a DYNAMIC parameter.
```

Example 1

```
$ WRITESYS$OUTPUT F$GETSYI(“USER3”)
$ USERD1 = F$GETSYI(“USERD1”)
```

Example 2

Controlling System Startup

There are times during system maintenance when the options OpenVMS provides for system startup (STARTUP_Pn parameters and SET/STARTUP) are limited in their functionality. Incorporating the SYSGEN USER parameters into a startup procedure can increase system startup flexibility and allow a system manager to complete maintenance operations in less time. **Example 3** shows the USER3 parameter being used to control which parts of SYSTARTUP_VMS.COM get executed. In this example, if USER3 equals 1, then only the network stacks will be started and all the disks mounted. None of the application startup procedures will be executed and DECWindows will not be started.

```
$ IF F$GETSYI(“USER3”) .EQ. 1 THEN GOTO No_Application_Boot
$ ! Normal startup procedure would follow here
.
.
$ EXIT 1    ! End of normal startup procedures
$ No_Application_Boot:
$ START/NETWORK DECNET
$ @SYS$MANAGER:TCPIP$STARTUP
$ @SYS$MANAGER:MOUNT_DISKS.COM
$ DEFINE DECW$IGNORE_DECWINDOWS TRUE
$ STARTUP$INTERACTIVE_LOGINS :== 0
$ EXIT 1
```

Example 3

This is a fairly simple example, but a system manager could implement many different startup configurations all based on the values to which the USER parameters are set. One USER parameter could be used to control which network ports are configured into the system while another USER parameter could be used to control whether a system would participate as a server for a TCPIP-based service. Adding these parameters into your system startup command procedures can significantly enhance the maintainability of a system.

Controlling Application Startup

There are times when a system manager would like to boot a system and start up the system applications in a non-default manner. The USER parameters provide an excellent way to do this. You can edit SYLOGICALS.COM to employ the USER parameters to determine to which values the application logicals should be set. In a large cluster this strategy allows a system manager to have one subset of nodes act as the main application servers and another subset act as backup servers --- or the USER parameters can be used to disable individual parts of an application. These parameters can also be used to enable database verification procedures before normal application startup. The possibilities are limited only by how much control the application startup procedures provide.

Documenting USER Usage

The last -- and very important! -- step in implementing the USER parameters is to document how they have been employed. The best place to document their usage in a system or a cluster is directly in the file where they are used, whether it be MODPARAMS.DAT, SYSTARTUP_VMS.COM, or any other command procedure. You should also print and keep a hardcopy document with other system documentation such as boot profiles and system configuration profiles.