HFRD Guide to the CONFGR Utility

28th May 1992

A guide to HFRD configuration control of VAXcluster startup, AUTOGEN and DECwindows server customization files using the CONFGR utility.

This guide assumes a knowledge of the purpose and use of VMS system management facilities.

Contact: M.G.Daniel

Printed Copy: This book is available for printing to a PostScript printer. At the VMS command line enter ‘‘PrintBook’’, the facility is menu driven. For multiple copies it is better to print once, then *photo-copy*.

This document was prepared using VAX DOCUMENT v2.1

Chapter 1
Introduction

HFRD's VAXcluster is very large by any standard, presently some 60 systems. These range from NI satellite VAXstation 2000s, with no internal disks and 14 Mb of memory, to a CI connected VAX 9000-420 with 256Mb of memory, and serving disks to the satellite systems. The cluster has 10 system disks.

The implications of such a large and disparate cluster stretch the standard methods DEC uses for system startup and AUTOGEN control. HFRD VMS Systems Management has adapted these to provide less cumbersome methods, as well as a system for much greater understanding and control of the cluster on a macro and individual system level.

The paradigm used is ''centralized configuration control'', managing startup, AUTOGEN and DECwindows setup files for each of the individual systems from a single configuration file for each aspect. This file specifies how each system should be configured, and a utility generates the required system files in a central directory, before actually introducing them into production, this occuring only after the utility is used to copy the generated files to system specific directories.

1.1 Startup Control

Perceived problems with the standard DEC method as applied to large VAXclusters:

- There are two general approaches for VMS startup control in a cluster:
  1. Individual, system-specific SYS$SYSROOT:[SYSMGR]SYSTARTUP_V5.COM
  2. A cluster-common SYSTARTUP_V5.COM file
- When using system-specific startup files, a small change required for all nodes or to one aspect of startup in a cluster must be individually applied in each case. This is fine for one or two systems, but once the cluster grows to a substantial number of systems it rapidly becomes unmanageable.

- The alternative, a cluster-common SYSTARTUP_V5.COM file, soon becomes full of *if-then-elses* in a cluster where members have differing management/application requirements. This is fine for one or two systems, but again once the cluster grows it becomes increasingly difficult to manage.

Both these methods tend to make centralized management difficult. It also renders it almost impossible for systems management staff to gain an overall appreciation of how a large cluster is configured without separate and extensive documentation which may or may not faithfully represent the actual or current situation (a problem with all external documentation!)

1.2 SHUTDOWN Control

- There can be two approaches in VMS system-specific shutdown in a cluster:
  1. Individual, system-specific SYS$SYSROOT:[SYSMGR]SYSHUTDWN.COM
  2. System-disk-common SYS$COMMON:[SYSMGR]SYSHUTDWN.COM
- As with startup control, when using system-specific startup files, a small change required for all nodes or to one aspect of shutdown in a cluster must be individually applied in each case. Once the cluster grows to a substantial number of systems this becomes unmanageable.
- The alternative, a common SYSHUTDWN.COM file is only system-disk common. this is becomes more complicated in a cluster containing multiple system disks. It also must contain *if-then-elses* in a cluster where members have differing shutdown requirements. This is fine for one or two systems, but as the cluster grows it becomes increasingly difficult to manage.

Not unlike startups both these alternatives tend to make centralized management difficult.

1.3 AUTOGEN Control

The AUTOGEN utility is used to ‘‘tune’’ VMS systems based on data collected from a running system along with guidance directives provided by systems management allowing custom SYStem GENeration and tuning. Perceived problems with the standard DEC method as applied to large VAXclusters:

- Each of the directive files, SYS$SYSROOT:[SYSEXE]MODPARAMS.DAT, must be individually maintained to provide both cluster and system specific parameter control. A small change required for all nodes in a cluster must be applied to each file. This is fine for one or two systems, but once the cluster grows to a substantial number of systems it quickly becomes unmanageable.
- Each of the system-specific parameter files is necessarily in a system- specific location, SYS$SYSROOT:[SYSEXE]. As with startups, this tends to make centralized management more difficult and makes it almost impossible for systems management staff to gain an overall appreciation of how a large cluster is configured.

1.4 DECwindows Server Control

The site-configurable DECwindows server setup procedure, DECW$PRIVATE_SERVER_SETUP.COM, allows systems management to customize the DECwindows X-server startup. This includes providing DCL SYMBOLS to control such things as monitor density and server process quotas.

These server setup procedures must be located on each system disk in either the SYS$COMMON:[SYSMGR] or SYS$SPECIFIC:[SYSMGR] directories.

Perceived problems with the standard DEC method as applied to large VAXclusters:

- If located in the SYS$COMMON area the procedure must contain decision structures and separate setup sections for each system being customized. This works well enough for a small number of systems, but once the cluster grows it becomes more difficlt to manage.
- If located in the SYS$SPECIFIC area each of the system-specific procedures must be individually maintained. As with startups, this tends to make centralized management more difficult and makes it almost impossible for systems management staff to gain an overall appreciation of how a large cluster is configured.

Chapter 2
CONFGR Overview

CONFGR is an HFRD implemented utility.

The approach used is ‘‘centralized configuration control’’, managing startup and AUTOGEN files for each of the individual systems from a single configuration file for each aspect.

This file specifies how each system should be configured, and the CONFGR utility generates the required system files in a central directory.

The generated files are then placed into production as required, using the same utility to locate a system’s specific directory and then copy them into this.

This approach has the following benefits:

1. The configuration file provides a natural, accurate and succinct overview of how a large number of separate aspects are used by systems according to individual requirements.
2. Changes to an specific aspect can rapidly, accurately and easily be propagated to all affected individual systems.
3. Separates the generation of individual system files from the placing of these into production. This allows the checking of changes, etc., before committing systems to them.

Chapter 3
Startup

Effective use of the CONFGR utility to manage startups depends on the two aspects, **configuration control** and **generation and distribution**.

Configuration Control

Configuration control consists of:

- Grouping startup-related DCL commands according to activity into files named in a meaningful way. This is where care and consideration in breaking startup into useful groupings must be exercised. **Judicious use of comments within each DCL file is essential.**

  This activity comprises creating and/or modifying the CONFGR$$STARTUP:*activity*.COM files.

- Maintaining the overall configuration. Each system has its own entry comprising one or more lines, with comma-separated fields specifying the names of the startup DCL files.

  Note: The SYSTARTUP_V5.COM file is generated in the same order as the system's specification in the configuration file. Hence startup functions that must be done before or after others must be specified in that order.

  This activity comprises the editing the CONFGR$$STARTUP:CONFGR.DAT file.

Generation and Distribution

As the Generation and distribution activities are separate, CONFGR may be used to check the configuration file and startup DCL files without affecting any system's actual SYSTARTUP_V5.COM. This central collection of system-specific SYSTARTUP_V5.COM files allows ready inspection of a system's startup without being on that node or accessing it's system root.

The generation and distribution activities are handled by the CONFGR utility.

Note

Generally, where an application installation indicates that the startup should be placed in or modifications made to SYSTARTUP_V5.COM then make them using the CONFGR utility.

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSMGR]SYSTARTUP_V5.COM file, use the CONFGR utility.

3.1 Startup DCL Files

Startup DCL files are basically standalone DCL procedures. These provide the DCL to perform the startup activity. A startup activity may just be a call to an application startup procedure located elsewhere, or can be simple or complex stand-alone DCL procedures.

Simple startup DCL file:

```
$!------------------------------------------------------------------------------
$! PASCAL.COM
$! Provide DEC VAX Pascal.
$!
$! 22-JUN-91  MGD  INITIAL DEVELOPMENT
$!------------------------------------------------------------------------------
$ DEFINE SITE$NODE$APPLICATIONS /SYSTEM /NOLOG -
         "''F$TRNLNM("SITE$NODE$APPLICATIONS")'Pascal!"
$ @SYS$STARTUP:PASCAL$STARTUP.COM
$!------------------------------------------------------------------------------
```

Although CONFGR$$STARTUP:PASCAL.COM may seem a trivial startup file, the use of the startup DCL names in the configuration file offers a self-documenting and obvious description of exactly what the system startup is all about (provided reasonably obvious and self-explanatory file names are chosen!).

More complex startup DCL file:

```
$!------------------------------------------------------------------------------
$! LAT.COM
$!
$! 12-APR-92  MGD  modified for use with STARTMAN
$! 23-APR-89  MGD  initial development (as LTLOAD.COM)
$!------------------------------------------------------------------------------
$!
$ HARDWARE = F$GETSYI("HW_NAME")
$ IF F$LOCATE("VAXstation",HARDWARE) .LT. F$LENGTH(HARDWARE)
$ THEN
$    HARDWARE = "VAXstation"
$ ELSE
$    IF F$LOCATE("VAXserver",HARDWARE) .LT. F$LENGTH(HARDWARE) -
       THEN HARDWARE = "VAXserver"
$ ENDIF
$!
$ RUN SYS$SYSTEM:SYSGEN
      CONNECT LTA0 /NOADAPTER
$!
$ LATCP = "$SYS$SYSTEM:LATCP"
$!
$ LATCP SET NODE /IDENT="High Frequency Radar Division ''HARDWARE'" /NOLOG
$!
$ LATCP CREATE SERVICE /IDENT="High Frequency Radar Division ''HARDWARE'" /NOLOG
$!
$ LATCP START NODE
$!------------------------------------------------------------------------------
```

Stand-alone DCL

It must be possible to interactively execute these DCL procedures standalone and have them accomplish their activity, i.e. they must be self-contained, with all the DCL required to execute successfully. No assumptions can be made about any other DCL code in the generated SYSTARTUP_V5.COM (apart from the presence of a ‘‘SAY’’ foreign verb assigned as ‘‘WRITE SYS$OUTPUT’’).

This allows the checking of startup DCL outside of an actual system startup, as well as the interactive starting or restarting of an application using a ‘‘@CONFGR$$STARTUP:*activity*.COM’’

All non-comment lines are simply read from the startup DCL file by the CONFGR utility and included in the systems’s specific SYSTARTUP_V5.COM

3.2 CONFGR.DAT File

CONFGR$$STARTUP:CONFGR.DAT provides the CONFGR utility with the overall configuration of each node.

Each system has its own entry, beginning with a quote-delimited system name (the SCSNODE name) and comprising one or more lines, with comma-separated fields specifying the names of the activity-specific startup DCL files. A system specification is considered continued until the next occurance of a quote-delimited system name. These entries should be similarly ordered for all systems.

Note

The SYSTARTUP_V5.COM file DCL is generated in the same order as the specifications in the configuration file. Hence startup functions that must be done in a specific order MUST be specified in that order.

An example CONFGR.DAT file:

```
   .
   .
   .
!
"DBOOT", secaudit, accounting, DECnet, LAT, DECwindows,
         queue_manager, initialize_batch_queues,
         Bookreader, Email,
         !-----
         Project_Manager, Pathworks, CDROM_software,
         !-----
         start_batch_queues, banners, finalize
!
   .
   .
   .
!
"WS48",  secaudit, accounting, DECnet, LAT, DECwindows,
         queue_manager, initialize_batch_queues,
         Bookreader, Email,
         !-----
         FORTRAN, Pascal, WordPerfect,
         !-----
         WS48_specific_startup, start_batch_queues, banners, finalize
!
   .
   .
   .
```

3.3 Generated SYSTARTUP_V5.COM File

All non-comment lines are simply read from the startup DCL files by the CONFGR utility and included in the systems's specific SYSTARTUP_V5.COM

Non-quoted text is by default changed to lower-case (the author considers this format easier to read), although this is optional.

The DCL included from each startup file is prefixed by a comment line including the name of the source startup DCL file. This makes the generated file easier to check.

The SYSTARTUP_V5.COM procedure provides informational messages concerning the stage of startup to the console of the system during startup. This provides users with startup progress information in the case of workstations, information also valuable when determining in which section of a startup problems occured.

Should a startup DCL file name be specified that does not exist DCL code is placed in the generated SYSTARTUP_V5.COM to display a message on the console indicating the problem, as well as a comment line message indicating the non- included file name. If any errors of this nature occured during a system's startup generation, DCL code is generated to display another warning message on the console at the conclusion of startup.

All non-comment lines are simply read from the startup DCL file by the CONFGR utility and included in the systems's specific SYSTARTUP_V5.COM. For satellite systems in particular this in-line DCL code is considerably more efficient than a collection of ''@'' execution commands, especially for activities comprising only a small number of commands.

Of course, the CONFGR utility does not assess the correctness of the DCL specified in the startup files!

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSMGR]SYSTARTUP_V5.COM file, use the CONFGR utility.

Example Generated SYSTARTUP_V5.COM

```
$! SYSTARTUP_V5.COM for DEFAULT
$! Created: Monday, 17-AUG-1992 08:49
$! Produced by: $1$dua8:[site_manager]confgr.exe
$!
$! DO NOT MODIFY this file directly, use the CONFGR utility!
$!
$ set noon
$ say = "WRITE SYS$OUTPUT"
$ say ""
$ say "%SYSTARTUPV5-I-BEGIN, system specific startup for DEFAULT"
$!
$!------------------------- INITIALIZE ------------------------------!
$ say ""
$ say "%SYSTARTUPV5-I-NOW, ""initialize"""
$ nodename = f$getsyi("NODENAME")
$ define /system site$access_shortly "TRUE"
$ set accounting /disable
$ if f$locate("VAXstation",f$getsyi("HW_NAME")) .eq. 0 -
     then set accounting /disable
$ define /nolog /system /exec shutdown$inform_nodes "''nodename'"
$!
$!------------------------- SECAUDIT --------------------------------!
$ say ""
$ say "%SYSTARTUPV5-I-NOW, ""secaudit"""
$ audit_file = "''f$trnlnm("CCDB$$DIR")'SECURITY_AUDIT.AUDIT$JOURNAL"
$ set audit /journal -
       /destination='audit_file' -
       /server=redirect_system_log
$ set audit /journal -
      /server=final_action=ignore_new -
      /resource=monitor_mode=percentage
$ set audit /alarm -
      /enable=(authorization, -
               acl, -
               login=(all), -
               logfail=(all), -
               breakin=(all)) -
      /disable=(login=(subprocess))
$!
$!------------------------- DEFAULT ---------------------------------!
$ say ""
$ say "%SYSTARTUPV5-I-NOW, ""default"""
$ say ""
$ say ""
```

```
$ say "*****************************************************************************"
$ say "*                                                                           *"
$ say "*  THIS NODE HAS NOT HAD A confgr GENERATED ""SYSTARTUP_V5.COM"" PROVIDED  * "
$ say "*                                                                           *"
$ say "*       the startup will only be partial, please correct immediately        *"
$ say "*                                                                           *"
$ say "*****************************************************************************"
$ say ""
$ say ""
$!
$!------------------------- DECNET ----------------------------------!
$ say ""
$ say "%SYSTARTUPV5-I-NOW, ""DECnet"""
$ if f$locate("VAXstation",f$getsyi("HW_NAME")) .eq. 0
$ then
$    say "%DECNET-I-HFRD, setting VAXstation requirements"
$    run sys$system:ncp
     define executor maximum broadcast nonrouters 256
     define known line receive buffer 24
     define exec pipeline 24000
     define exec alias node vaxc
     define exec alias incoming disable
$ else
$    say "%DECNET-I-HFRD, setting VAX requirements"
$    run sys$system:ncp
     define executor maximum broadcast nonrouters 256
     define known line receive buffer 32
     define exec pipeline 32000
     define exec alias node vaxc
     define exec alias incoming disable
$ endif
$ nodename = f$getsyi("NODENAME")
$ if nodename .eqs. "DAJAV" .or. nodename .eqs. "DAMSC" .or. -
     nodename .eqs. "DBOOT" .or. nodename .eqs. "SERV2"
$ then
$    say "%DECNET-I-HFRD, setting incoming alias enabled"
$    run sys$system:ncp
     define exec alias node vaxc
     define exec alias incoming enable
$ endif
$ say "%DECNET-I-HFRD, starting DECnet"
$ @sys$manager:startnet.com
$!
$!------------------------- LAT -------------------------------------!
$ say ""
$ say "%SYSTARTUPV5-I-NOW, ""LAT"""
$ hardware = f$getsyi("HW_NAME")
$ if f$locate("VAXstation",hardware) .lt. f$length(hardware)
$ then
$    hardware = "VAXstation"
$ else
$    if f$locate("VAXserver",hardware) .lt. f$length(hardware) -
        then hardware = "VAXserver"
$ endif
$ run sys$system:sysgen
      connect lta0 /noadapter
$ latcp = "$SYS$SYSTEM:LATCP"
$ latcp set node /ident="High Frequency Radar Division ''hardware'" /nolog
$ latcp create service /ident="High Frequency Radar Division ''hardware'" /nolog
```

```
$ latcp start node
$!
$ say ""
$ say "%SYSTARTUPV5-I-END, system specific startup for DEFAULT"
$ say ""
```

3.4 Using CONFGR

There are 4 basic steps when using the CONFGR utility.

1. Create or modify any startup DCL files, as required.
2. Edit the CONFGR.DAT configuration file, as required.
3. Generate the system-specific SYSTARTUP_V5.COM file.

   This will be created in the CONFGR$$STARTUP directory and named SYSTARTUP_V5.*system-name*.

   The CONFGR command for this activity is:
   ''CONFGR /STARTUP [/NODE=*name*[,*name*]]''

   This has not altered any system's actual SYSTARTUP_V5.COM. It is possible at this stage to review, change, delete, etc., any of the configuration files.
4. Distribute the generated SYSTARTUP_V5.COM files to system-specific directories.

   This copies the SYSTARTUP_V5.*system-name* file, from the CONFGR$$STARTUP directory to the system's specific root, *disk*:[*sysroot*.SYSMGR]. It is renamed from SYSTARTUP_V5.*system-name* to SYSTARTUP_V5.COM.

   The CONFGR command for this activity is:
   ''CONFGR /STARTUP /COPY [/NODE=*name*[,*name*]]''

3.4.1 Example sessions

Generating SYSTARTUP_V5.COM Files

```
DAJAV$ confgr /startup /node=ws1,ws2,ws3,ws4
Configure SYSTARTUP_V5.COM for WS1? [Y]:
%CONFGR-I-NOW, SYSTARTUP_V5.COM for WS1
Configure SYSTARTUP_V5.COM for WS2? [Y]:
%CONFGR-I-NOW, SYSTARTUP_V5.COM for WS2
Configure SYSTARTUP_V5.COM for WS3? [Y]: n
Configure SYSTARTUP_V5.COM for WS4? [Y]:
%CONFGR-I-NOW, SYSTARTUP_V5.COM for WS4
%CONFGR-I-CONFGRED, WS1,WS2,WS4
%CONFGR-W-NOTCONFGRED, WS3
DAJAV$
```

Distributing SYSTARTUP_V5.COM Files

```
DAJAV$ confgr /startup /node=ws1,ws2,ws3,ws4 /copy
%CONFGR-I-NOW, checking system roots on $1$DUA4:
%CONFGR-I-NOW, checking system roots on $1$DUA5:
%CONFGR-I-NOW, checking system roots on $1$DUA6:
%CONFGR-I-NOW, checking system roots on $1$DUA7:
%CONFGR-I-NOW, checking system roots on $1$DUA8:
%CONFGR-I-NOW, checking system roots on $1$DUA9:
%CONFGR-I-NOW, checking system roots on $1$DUA10:
%CONFGR-I-NOW, checking system roots on $1$DUA11:
%CONFGR-I-NOW, checking system roots on $1$DUA16:
%CONFGR-I-NOW, checking system roots on $1$DUA17:
%CONFGR-I-NOW, checking system roots on $1$DUA22:
%CONFGR-I-NOW, checking system roots on $1$DUA23:
%CONFGR-I-NOW, checking system roots on $1$DUA24:
%CONFGR-I-NOW, checking system roots on $1$DUA25:
%CONFGR-I-NOW, checking system roots on $1$DUA34:
%CONFGR-I-NOW, checking system roots on $1$DUA35:
%CONFGR-I-NOW, checking system roots on $1$DUA36:
%CONFGR-I-NOW, checking system roots on $1$DUA37:
%CONFGR-I-NOW, checking system roots on $1$DUA0:
%CONFGR-I-NOW, checking system roots on $1$DUA1:
%CONFGR-I-NOW, checking system roots on $1$DUA2:
%CONFGR-I-NOW, checking system roots on $1$DUA12:
%CONFGR-I-NOW, checking system roots on $1$DUA13:
%CONFGR-I-NOW, checking system roots on $1$DUA14:
%CONFGR-I-NOW, checking system roots on $1$DUA15:
%CONFGR-I-NOW, checking system roots on $1$DUA18:
%CONFGR-I-NOW, checking system roots on $1$DUA19:
%CONFGR-I-NOW, checking system roots on $1$DUA20:
%CONFGR-I-NOW, checking system roots on $1$DUA21:
%CONFGR-I-NOW, checking system roots on $1$DUA30:
%CONFGR-I-NOW, checking system roots on $1$DUA31:
%CONFGR-I-NOW, checking system roots on $1$DUA32:
%CONFGR-I-NOW, checking system roots on $1$DUA33:
%CONFGR-I-NOW, checking system roots on SERV2$DUA0:
%CONFGR-I-NOW, checking system roots on SERV2$DUA1:
%CONFGR-I-NOW, checking system roots on SERV2$DUA2:
%CONFGR-I-NOW, checking system roots on SERV2$DUA3:
%CONFGR-I-NOW, checking system roots on SERV2$DUB255:
%CONFGR-I-NOW, checking system roots on DBOOT$DUA0:
%CONFGR-I-NOW, checking system roots on DBOOT$DUA4:
Copy SYSTARTUP_V5.COM for WS3 to DBOOT$DUA4:[SYS13.SYSMGR]? [Y]: n
%CONFGR-I-NOW, checking system roots on DBOOT$DUA5:
Copy SYSTARTUP_V5.COM for WS1 to DBOOT$DUA5:[SYS10.SYSMGR]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$STARTUP]SYSTARTUP_V5.WS1;63 copied to D
%PURGE-I-FILPURG, DBOOT$DUA5:[SYS10.SYSMGR]SYSTARTUP_V5.COM;13 deleted (27 blocks)
Copy SYSTARTUP_V5.COM for WS2 to DBOOT$DUA5:[SYS18.SYSMGR]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$STARTUP]SYSTARTUP_V5.WS2;62 copied to D
%PURGE-I-FILPURG, DBOOT$DUA5:[SYS18.SYSMGR]SYSTARTUP_V5.COM;24 deleted (27 blocks)
%CONFGR-I-NOW, checking system roots on DBOOT$DUA6:
Copy SYSTARTUP_V5.COM for WS4 to DBOOT$DUA6:[SYS15.SYSMGR]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$STARTUP]SYSTARTUP_V5.WS4;62 copied to D
%PURGE-I-FILPURG, DBOOT$DUA6:[SYS15.SYSMGR]SYSTARTUP_V5.COM;53 deleted (27 blocks)
%CONFGR-I-NOW, checking system roots on DBOOT$DUB255:
%CONFGR-I-CONFGRED, WS1,WS2,WS4
%CONFGR-W-NOTCONFGRED, WS3
DAJAV$
```

Chapter 4
Shutdown

The management of system-specific shutdowns using CONFGR is very similar to that of startups, so example files and example usage is not included in the section. This management depends on the two aspects, **configuration control** and **generation and distribution**.

Configuration Control

Configuration control consists of:

- Grouping shutdown-related DCL commands according to activity into files named in a meaningful way. **Use of comments within each DCL file is essential.**

  This activity comprises creating and/or modifying the CONFGR$$SHUTDOWN:*activity*.COM files.

- Maintaining the overall configuration. Each system has its own entry comprising one or more lines, with comma-separated fields specifying the names of the shutdown DCL files.

  Note: The SYSHUTDWN.COM file is generated in the same order as the system's specification in the configuration file. Hence shutdown functions that must be done before or after others must be specified in that order.

  This activity comprises the editing the CONFGR$$SHUTDOWN:CONFGR.DAT file.

Generation and Distribution

The generation and distribution activities are handled by the CONFGR utility.

Note

Generally, where an application installation indicates that the shutdown should be placed in or modifications made to SYSHUTDWN.COM then make them using the CONFGR utility.

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSMGR]SYSHUTDWN.COM file, use the CONFGR utility.

4.1 Shutdown DCL Files

Shutdown DCL files are basically standalone DCL procedures. These provide the DCL to perform the shutdown activity. A shutdown activity may just be a call to an application shutdown procedure located elsewhere, or can be simple or complex stand-alone DCL procedures. They have the same role as those used for startup configuration.

Stand-alone DCL

It must be possible to interactively execute these DCL procedures standalone and have them accomplish their activity, i.e. they must be self-contained, with all the DCL required to execute successfully. No assumptions can be made about any other DCL code in the generated SYSHUTDWN.COM (apart from the presence of a ''SAY'' foreign verb assigned as ''WRITE SYS$OUTPUT'').

All non-comment lines are simply read from the shutdown DCL file by the CONFGR utility and included in the systems's specific SYSHUTDWN.COM

4.2 CONFGR.DAT File

CONFGR$$SHUTDOWN:CONFGR.DAT provides the CONFGR utility with the overall configuration of each node.

Each system has its own entry, beginning with a quote-delimited system name (the SCSNODE name) and comprising one or more lines, with comma-separated fields specifying the names of the activity-specific shutdown DCL files. A system specification is considered continued until the next occurance of a quote-delimited system name.

Note

The SYSHUTDWN.COM file DCL is generated in the same order as the specifications in the configuration file. Hence shutdown functions that must be done in a specific order MUST be specified in that order.

4.3 Generated SYSHUTDWN.COM File

All non-comment lines are simply read from the shutdown DCL files by the CONFGR utility and included in the systems's specific SYSHUTDWN.COM

Non-quoted text is by default changed to lower-case (the author considers this format easier to read), although this is optional.

The DCL included from each shutdown file is prefixed by a comment line including the name of the source shutdown DCL file. This makes the generated file easier to check.

The SYSHUTDWN.COM procedure provides informational messages concerning the stage of shutdown to the console of the system during shutdown. This provides users with shutdown progress information in the case of workstations, information also valuable when determining in which section of a shutdown problems occured.

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSMGR]SYSHUTDWN.COM file, use the CONFGR utility.

4.4 Using CONFGR

There are 4 basic steps when using the CONFGR utility.

1. Create or modify any shutdown DCL files, as required.
2. Edit the CONFGR.DAT configuration file, as required.
3. Generate the system-specific SYSHUTDWN.COM file.

   This will be created in the CONFGR$$SHUTDOWN directory and named SYSHUTDWN.*system-name*.

   The CONFGR command for this activity is:
   ''CONFGR /SHUTDOWN [/NODE=*name*[,*name*]]''

   This has not altered any system's actual SYSHUTDWN.COM. It is possible at this stage to review, change, delete, etc., any of the configuration files.
4. Distribute the generated SYSHUTDWN.COM files to system-specific directories.

   This copies the SYSHUTDWN.*system-name* file, from the CONFGR$$SHUTDOWN directory to the system's specific root, *disk*:[*sysroot*.SYSMGR]. It is renamed from SYSHUTDWN.*system-name* to SYSHUTDWN.COM.

   The CONFGR command for this activity is:
   ''CONFGR /SHUTDOWN /COPY [/NODE=*name*[,*name*]]''

Chapter 5
Modparams/AUTOGEN

Effective use of the CONFGR utility to manage AUTOGEN MODPARAMS.DAT files depends on the two aspects, **configuration control** and **generation and distribution**.

Configuration Control

Configuration control consists of:

- Grouping related SYSGEN parameters and AUTOGEN directives according to purpose into parameter files named in a meaningful way. This is where care and consideration in breaking these parameters into meaningful groupings must be exercised. **Judicious use of comments within each parameter file is essential.**

  This activity comprises creating and/or modifying the CONFGR$$MODPARAMS:*group*.PAR files.

- Maintaining the overall configuration. Each system has its own entry comprising one or more lines, with comma-separated fields specifying the names of the parameter files.

  This activity comprises the editing the CONFGR$$MODPARAMS:CONFGR.PAR file.

Generation and Distribution

As the Generation and distribution activities are separate, CONFGR may be used to check the both configuration and parameter files without affecting any system's actual MODPARAMS.DAT. This central collection of system-specific MODPARAMS files allows ready inspection of a system's AUTOGEN directive file without being on that node or accessing it's system root.

The generation and distribution activities are handled by the CONFGR utility.

Note

Generally, where an application installation indicates that the shutdown should be placed in or modifications made to MODPARAMS.DAT then make them using the CONFGR utility.

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT file, use the CONFGR utility.

Consistency Checking

No checking of generated MODPARAMS.DAT parameters is done by the CONFGR utility.

It is therefore not only possible but highly likely that the parameters will interact. That is, there will be multiple ''add_''s, ''min_''s, etc. affecting the same SYSGEN parameters. As AUTOGEN does its own consistency checking this should not present problems.

However, for the user to be aware of exactly how the generated parameters are interacting it is advisable to check the SYS$SYSTEM:AGEN$PARAMS.REPORT files for comments made by AUTOGEN about these effects.

5.1 Parameter Files

Parameter files are data files containing meaningfully grouped SYSGEN parameters or AUTOGEN directives (''min_'', etc.). Grouping may be made according to system type, application requirements, or functional specification such as a system being a voting member.

Example system-type parameter file:

```
!-------------------------------------------------------------------------------
! VAX3100.PAR
! VAXstation 3100s
!
! 24-JUL-92  MGD  POSIX: maxprocesscnt minimum now 60, add 10 for good measure
! 16-MAY-92  MGD  disable tape MSCP serving for VAXstations
! 29-APR-92  MGD  Workstation replay, > 24000 VIRTUALPAGECNT (default at 8Mb)
! 28-APR-92  MGD  ACP_REBLDSYSD on satellites to inhibit system disk rebuilds
! 02-APR-92  MGD  "show mem /pool /full" indicates increased SRP, IRP, LRP
! 12-SEP-91  MGD  changed "lockdirwt" to 0
! 07-MAY-91  MGD  initial development
!-------------------------------------------------------------------------------
!
! "lockdirwt" in proportion to CPU power, VS3100 approx 2.7 VUP
! "lockdirwt" for non-MSCP serving nodes should be 0
LOCKDIRWT = 0
!
! based on comparing "initial" with "current" values for SRP, IRP, LRP
! using "show mem /pool /full", increase "min_" to a bit more than "current"
MIN_SRPCOUNT=4000
MIN_IRPCOUNT=2000
MIN_LRPCOUNT=150
!
ADD_GBLPAGES=2000
ADD_GBLSECTIONS=3
!
MIN_MAXPROCESSCNT=60
ADD_MAXPROCESSCNT=10
!
MIN_VIRTUALPAGECNT=40000
!
ACP_REBLDSYSD=0
```

```
!
TMSCP_LOAD = 0
!
!-------------------------------------------------------------------------------
```

Example functional parameter file:

```
!-------------------------------------------------------------------------------
! VOTES_1.PAR
! Voting nodes contribute one vote each (apart from DAJAV).
!
! 17-APR-92  MGD  due to the arrangement of system disks and to ensure an odd
!                 number of votes DAJAV has been given two votes, with DAMSC,
!                 DARSC, DADTC each with 1 , contributing another 3.
! 07-MAY-91  MGD  initial development
!-------------------------------------------------------------------------------
VOTES = 1
!-------------------------------------------------------------------------------
```

Although CONFGR$$MODPARAMS:VOTES-1.PAR may seem a trivial parameter file, the use of the names in the configuration file offers a self-documenting and obvious description of exactly what the system SYSGEN specification is all about (provided reasonably obvious and self-explanatory file names are chosen!).

All non-comment lines are simply read from the parameter by the CONFGR utility and included in the systems's specific MODPARAMS.DAT

5.2 CONFGR.PAR File

CONFGR$$MODPARAMS:CONFGR.PAR provides the CONFGR utility with the overall AUTOGEN configuration of each node.

Each system has its own entry, beginning with a quote-delimited ''SCSNODE'' system name, the ''SCSSYSTEMID'' number and then comprising one or more lines, with comma-separated fields specifying the names of the parameter files. A system specification is considered continued until the next occurance of a quote-delimited system name. Entries should be similarly ordered for all systems.

An example CONFGR.PAR file:

```
!-------------------------------------------------------------------------
! CONFGR.PAR
! master node parameter file for vaxc
!
! 07-MAY-91  MGD  initial development
!-------------------------------------------------------------------------
!
"DAJAV", 1530, VAXC, VOTES-2, HSC-DISKSERVER, VAX9420, PATHWORKS,
         POSIX, MULTINET
"DAMSC", 1527, VAXC, VOTES-1, HSC-DISKSERVER, VAX6220, PATHWORKS,
         POSIX, EWS, TEAMWORK
"DARSC", 1525, VAXC, VOTES-1, NON-DISKSERVER, VAX8250, PATHWORKS,
         POSIX, DARSC_YATES
"DADTC", 1526, VAXC, VOTES-1, NON-DISKSERVER, VAX8250, PATHWORKS,
         POSIX
!
"DBOOT", 1534, VAXC, NON-VOTING, LOCAL-DISKSERVER, VAX3600, PATHWORKS,
         POSIX
"SERV2", 1614, VAXC, NON-VOTING, LOCAL-DISKSERVER, VAX3600, PATHWORKS,
         POSIX
!
"WS1", 1535, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
       POSIX, TEAMWORK
"WS2", 1536, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
       POSIX, TEAMWORK
"WS3", 1537, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
       POSIX, TEAMWORK
"WS4", 1538, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
       POSIX, TEAMWORK
"WS5", 1539, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
       POSIX, TEAMWORK
"WS6", 1540, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
       POSIX, TEAMWORK
"WS7", 1541, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
       POSIX, TEAMWORK
"WS8", 1542, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
       POSIX, TEAMWORK
"WS9", 1543, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
       POSIX, TEAMWORK
!
"WS10", 1544, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS11", 1545, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS12", 1546, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS13", 1547, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS14", 1548, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS15", 1549, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
! "WS16" has metamorphised into EWS1
"WS17", 1551, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK,
               RADARTASKING
"WS18", 1552, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS19", 1553, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
```

```
        POSIX, TEAMWORK
!
"WS20", 1554, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS21", 1555, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS22", 1556, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS23", 1557, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS24", 1558, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS25", 1559, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS26", 1560, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS27", 1561, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS28", 1562, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS29", 1563, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
!
"WS30", 1564, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
"WS31", 1565, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        FORCE_BUGCHECK_DUMP, POSIX, TEAMWORK
"WS32", 1566, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS33", 1567, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS34", 1568, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS35", 1569, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS36", 1570, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS37", 1571, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS38", 1572, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS39", 1573, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
!
"WS40", 1574, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS41", 1575, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS42", 1576, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS43", 1577, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK,
              WS43
"WS44", 1578, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS45", 1579, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS46", 1580, VAXC, NON-VOTING, NON-DISKSERVER, VAX2000, DECWINDOWS,
        POSIX, TEAMWORK
```

```
"WS47", 1581, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS48", 1582, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS,
        POSIX, TEAMWORK
"WS49", 1583, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
!
"WS50", 1584, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK,
               RADARTASKING
"WS51", 1585, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS52", 1586, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS53", 1587, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS54", 1588, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        FORCE_BUGCHECK_DUMP, POSIX, TEAMWORK
"WS55", 1589, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS56", 1590, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS57", 1591, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
"WS58", 1592, VAXC, NON-VOTING, NON-DISKSERVER, VAX4000, DECWINDOWS,
        POSIX, TEAMWORK
!
! (VCS is for when the "VAXcluster Console System" workstation occasionally)
! (joins the VAXC for system related activities, upgrades, etc.)
"VCS", 1615, VAXC, NON-VOTING, NON-DISKSERVER, VAX3100, DECWINDOWS, POSIX, VCS
!
!------------------------------------------------------------------------------
```

Note

Great care must be taken to correctly specify the SCSSYSTEMID. Unpredicatable conditions could result from AUTOGENING with incorrect SCSSYSTEMIDs. If a system does not have a SCSSYSTEMID in the CONFGR.PAR file an error is indicated and no MODPARAMS.DAT file is generated.

5.3 Generated MODPARAMS.DAT File

All non-comment lines are simply read from the parameter files by the CONFGR utility and included in the systems's specific MODPARAMS.DAT

Non-quoted text is by default changed to lower-case (the author considers this format easier to read), although this is optional.

The parameters included from each file are prefixed by a comment line including the name of the source parameter file. This makes the generated file easier to check.

Should a parameter file name be specified that does not exist the problem is indicated to the user and a comment line placed in the generated MODPARAMS.DAT indicating the non-included file name.

Of course, the CONFGR utility does not assess the correctness of the parameters specified!

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT file, use the CONFGR utility.

Example Generated MODPARAMS.DAT

```
! MODPARAMS.DAT for SERV2
! Created: Monday, 27-JUL-1992 08:11
! Produced by: $1$dua8:[site_manager]confgr.exe
!
! DO NOT MODIFY this file directly, use the CONFGR utility!
!
scsnode="SERV2"
scssystemid=1614
!------------------------ VAXC ----------------------------------!
disk_quorum = ""
qdskvotes = 1
qdskinterval = 10
expected_votes = 1
vaxcluster = 2
interconnect = "NICI"
niscs_conv_boot = 0
niscs_load_pea0 = 1
niscs_port_serv = 0
scsconncnt = 434
recnxinterval = 75
pagefile = 0
swapfile = 0
dumpfile = 0
mvtimeout = 64000
tty_typahdsz = 2048
tty_altypahd = 2048
dumpstyle = 1
savedump = 1
!------------------------ NON-VOTING ----------------------------!
votes=0
!------------------------ LOCAL-DISKSERVER ----------------------!
mscp_load = 1
mscp_serve_all = 2
alloclass=0
min_mscp_buffer = 1024
min_scsbuffcnt = 1024
!------------------------ VAX3600 -------------------------------!
dumpstyle=0
lockdirwt=3
min_lockidtbl = 60000
min_srpcount=10000
min_irpcount=6000
min_lrpcount=200
add_gblpages = 1000
add_gblsections = 10
min_wsmax = 10000
window_system = 0
!------------------------ PATHWORKS -----------------------------!
add_gblpages = 800
add_gblsections = 20
add_npagedyn = 310000
min_channelcnt = 450
min_maxbuf = 4000
```

```
min_wsmax = 10000
min_sptreq = 16000
!------------------------ POSIX ----------------------------------!
add_gblpages = 2000
add_gblsections = 30
load_sys_images = 1
add_sptreq = 1000
min_wsmax = 4200
```

5.4 Using CONFGR

There are 4 basic steps when using the CONFGR utility.

1. Create or modify any parameter files, as required.
2. Edit the CONFGR.PAR configuration file, as required.
3. Generate the system-specific MODPARAMS.DAT file.

   This will be created in the CONFGR$$MODPARAMS directory and named MODPARAMS.*system-name*.

   The CONFGR command for this activity is:
   ''CONFGR /MODPARAMS [/NODE=*name*[,*name*]]''

   This has not altered any system's actual MODPARAMS.DAT. It is possible at this stage to review, change, delete, etc., any of the configuration files.
4. Distribute the generated MODPARAMS.DAT files to system-specific directories.

   This copies the MODPARAMS.*system-name* file, from the CONFGR$$MODPARAMS directory to the system's specific root, *disk*:[*sysroot*.SYSEXE]. It is renamed from MODPARAMS.*system-name* to MODPARAMS.COM.

   The CONFGR command for this activity is:
   ''CONFGR /MODPARAMS /COPY [/NODE=*name*[,*name*]]''

5.4.1 Example sessions

Generating SYSHUTDWN.COM Files

```
DAJAV$ confgr /modparams /node=ws1,ws2,ws3,ws4
Configure MODPARAMS.DAT for WS1? [Y]: n
Configure MODPARAMS.DAT for WS2? [Y]:
%CONFGR-I-NOW, MODPARAMS.DAT for WS2
Configure MODPARAMS.DAT for WS3? [Y]:
%CONFGR-I-NOW, MODPARAMS.DAT for WS3
Configure MODPARAMS.DAT for WS4? [Y]:
%CONFGR-I-NOW, MODPARAMS.DAT for WS4
%CONFGR-I-CONFGRED, WS2,WS3,WS4
%CONFGR-W-NOTCONFGRED, WS1
```

Distributing SYSHUTDWN.COM Files

```
DAJAV$ confgr /modparams /node=ws1,ws2,ws3,ws4 /copy
%CONFGR-I-NOW, checking system roots on $1$DUA4:
%CONFGR-I-NOW, checking system roots on $1$DUA5:
%CONFGR-I-NOW, checking system roots on $1$DUA6:
%CONFGR-I-NOW, checking system roots on $1$DUA7:
%CONFGR-I-NOW, checking system roots on $1$DUA8:
%CONFGR-I-NOW, checking system roots on $1$DUA9:
%CONFGR-I-NOW, checking system roots on $1$DUA10:
%CONFGR-I-NOW, checking system roots on $1$DUA11:
%CONFGR-I-NOW, checking system roots on $1$DUA16:
%CONFGR-I-NOW, checking system roots on $1$DUA17:
%CONFGR-I-NOW, checking system roots on $1$DUA22:
%CONFGR-I-NOW, checking system roots on $1$DUA23:
%CONFGR-I-NOW, checking system roots on $1$DUA24:
%CONFGR-I-NOW, checking system roots on $1$DUA25:
%CONFGR-I-NOW, checking system roots on $1$DUA34:
%CONFGR-I-NOW, checking system roots on $1$DUA35:
%CONFGR-I-NOW, checking system roots on $1$DUA36:
%CONFGR-I-NOW, checking system roots on $1$DUA37:
%CONFGR-I-NOW, checking system roots on $1$DUA0:
%CONFGR-I-NOW, checking system roots on $1$DUA1:
%CONFGR-I-NOW, checking system roots on $1$DUA2:
%CONFGR-I-NOW, checking system roots on $1$DUA12:
%CONFGR-I-NOW, checking system roots on $1$DUA13:
%CONFGR-I-NOW, checking system roots on $1$DUA14:
%CONFGR-I-NOW, checking system roots on $1$DUA15:
%CONFGR-I-NOW, checking system roots on $1$DUA18:
%CONFGR-I-NOW, checking system roots on $1$DUA19:
%CONFGR-I-NOW, checking system roots on $1$DUA20:
%CONFGR-I-NOW, checking system roots on $1$DUA21:
%CONFGR-I-NOW, checking system roots on $1$DUA30:
%CONFGR-I-NOW, checking system roots on $1$DUA31:
%CONFGR-I-NOW, checking system roots on $1$DUA32:
%CONFGR-I-NOW, checking system roots on $1$DUA33:
%CONFGR-I-NOW, checking system roots on SERV2$DUA0:
%CONFGR-I-NOW, checking system roots on SERV2$DUA1:
%CONFGR-I-NOW, checking system roots on SERV2$DUA2:
%CONFGR-I-NOW, checking system roots on SERV2$DUA3:
%CONFGR-I-NOW, checking system roots on SERV2$DUB255:
%CONFGR-I-NOW, checking system roots on DBOOT$DUA0:
%CONFGR-I-NOW, checking system roots on DBOOT$DUA4:
Copy MODPARAMS.DAT for WS3 to DBOOT$DUA4:[SYS13.SYSEXE]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$MODPARAMS]MODPARAMS.WS3;22 copied to DB
%PURGE-I-FILPURG, DBOOT$DUA4:[SYS13.SYSEXE]MODPARAMS.DAT;11 deleted (3 blocks)
%CONFGR-I-NOW, checking system roots on DBOOT$DUA5:
Copy MODPARAMS.DAT for WS1 to DBOOT$DUA5:[SYS10.SYSEXE]? [Y]: n
Copy MODPARAMS.DAT for WS2 to DBOOT$DUA5:[SYS18.SYSEXE]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$MODPARAMS]MODPARAMS.WS2;24 copied to DB
%PURGE-I-FILPURG, DBOOT$DUA5:[SYS18.SYSEXE]MODPARAMS.DAT;11 deleted (3 blocks)
%CONFGR-I-NOW, checking system roots on DBOOT$DUA6:
Copy MODPARAMS.DAT for WS4 to DBOOT$DUA6:[SYS15.SYSEXE]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$MODPARAMS]MODPARAMS.WS4;22 copied to DB
%PURGE-I-FILPURG, DBOOT$DUA6:[SYS15.SYSEXE]MODPARAMS.DAT;10 deleted (3 blocks)
%CONFGR-I-NOW, checking system roots on DBOOT$DUB255:
%CONFGR-I-CONFGRED, WS2,WS3,WS4
%CONFGR-W-NOTCONFGRED, WS1
DAJAV$
```

Chapter 6
DECwindows

As with startups an effective use of the CONFGR utility to manage DECwidnows customization depends on the two aspects, **configuration control** and **generation and distribution**.

Configuration Control

Configuration control consists of:

- Grouping DCL commands according to requirement into files named in a meaningful way. This is where care and consideration in breaking startup into useful groupings must be exercised. **Judicious use of comments within each DCL file is essential.**

  This activity comprises creating and/or modifying the CONFGR$$DECWINDOWS:*requirement*.COM files.

- Maintaining the overall configuration. Each system has its own entry comprising one or more lines, with comma-separated fields specifying the names of the requirement DCL files.

  This activity comprises the editing the CONFGR$$DECWINDOWS:CONFGR.DECW file.

Generation and Distribution

As the Generation and distribution activities are separate, CONFGR may be used to check the configuration file and requirement DCL files without affecting any system's actual DECwindows setup. This central collection of system- specific customization files allows ready inspection of a system's configurations without being on that node or accessing it's system root.

The generation and distribution activities are handled by the CONFGR utility.

Note

Generally, where an documentation indicates that a modification should made to DECW$PRIVATE_SERVER_SETUP.COM then make them using the CONFGR utility.

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSMGR]DECW$PRIVATE_SERVER_SETUP.COM file, use the CONFGR utility.

6.1 DECwindows DCL Files

DECwindows server customization generally consists of defining a logical or assigning a symbol to provide guidance information to the X-server startup. The *DECwindows DCL files* are basically DCL procedures. These procedure merely collect together these directives into meaningful groupings.

DECwindows requirement DCL file:

```
!------------------------------------------------------------------------------
! 100DPI.COM
! Set DECwindows server startup for 100 dots-per-inch screens.
!
! 75dpi are also large, 19" screens on VAXstation 4000's (1280x1024)
! 100dpi are large, 19" screens on VAXstation 2000's and 3100's (1024x768)
! 100dpi are also small, 14" screens on VAXstation 4000's (1280x1024)
!
! 07-MAY-92  MGD  initial file
!------------------------------------------------------------------------------
$ DECW$SERVER_DENSITY == 100
!------------------------------------------------------------------------------
```

Although CONFGR$$DECWINDOWS:100DPI.COM may seem a trivial requirement file, the use of the DCL names in the configuration file offers a self-documenting and obvious description of exactly what the server startup is being configured for (provided reasonably obvious and self-explanatory file names are chosen!).

6.2 CONFGR.DECW File

CONFGR$$DECWINDOWS:CONFGR.DECW provides the CONFGR utility with the overall configuration of each node.

Each system has its own entry, beginning with a quote-delimited system name (the SCSNODE name) and comprising one or more lines, with comma-separated fields specifying the names of the requirement-specific DCL files. A system specification is considered continued until the next occurance of a quote- delimited system name. These entries should be similarly ordered for all systems.

The CONFGR.DECW file at time of writing:

```
!------------------------------------------------------------------------------
! CONFGR.DECW
! Configuration file for DECwindows private server setups.
!
! 07-MAY-92  MGD  initial file, extension of startup and modparams management
!------------------------------------------------------------------------------
!
! The default private setup should be placed in the SYS$COMMON:[SYSMGR]
! to output a warning message if a node has not had a CONFGR done for its
! DECwindows setup.
"DEFAULT", 75DPI, DEFAULT
!
! Nothing to setup for these nodes,
! but better have a file in sys$specific:[sysmgr] or the default
! in sys$common:[sysmgr] will output spurious warning messages.
"DAJAV"
"DAMSC"
"DADTC"
"DARSC"
"DBOOT"
"SERV2"
!
"WS1", 100DPI
"WS2", 100DPI
"WS3", 100DPI
"WS4", 100DPI
"WS5", 100DPI
"WS6", 100DPI
"WS7", 100DPI
"WS8", 100DPI
"WS9", 100DPI
!
"WS10", 100DPI
"WS11", 100DPI
"WS12", 100DPI
"WS13", 100DPI
"WS14", 100DPI
"WS15", 100DPI
! "WS16" has metamorphised into EWS1
!!"WS16", 100DPI
"WS17", 75DPI
"WS18", 75DPI
"WS19", 100DPI
!
"WS20", 100DPI
"WS21", 100DPI
"WS22", 100DPI
"WS23", 100DPI
"WS24", 100DPI
"WS25", 100DPI
"WS26", 100DPI
"WS27", 100DPI
"WS28", 100DPI
"WS29", 100DPI
"WS30", 100DPI
!
"WS31", 100DPI
"WS32", 75DPI
"WS33", 75DPI
```

```
"WS34", 75DPI
"WS35", 75DPI
"WS36", 75DPI
"WS37", 100DPI
"WS38", 100DPI
"WS39", 100DPI
!
"WS40", 100DPI
"WS41", 100DPI
"WS42", 75DPI
"WS43", 75DPI
"WS44", 100DPI
"WS45", 75DPI
"WS46", 75DPI
"WS47", 100DPI
"WS48", 75DPI
"WS49", 100DPI
!
"WS50", 100DPI
"WS51", 100DPI
"WS52", 100DPI
"WS53", 100DPI
"WS54", 100DPI
"WS55", 100DPI
"WS56", 100DPI
"WS57", 100DPI
"WS58", 100DPI
!
"VCS", 100DPI
!
!------------------------------------------------------------------------------
```

6.3 Generated DECW$PRIVATE_SERVER_SETUP.COM File

All non-comment lines are simply read from the requirement DCL files by the CONFGR utility and included in the systems's specific DECW$PRIVATE_SERVER_SETUP.COM

Non-quoted text is by default changed to lower-case (the author considers this format easier to read), although this is optional.

The DCL included from each file is prefixed by a comment line including the name of the source DCL file. This makes the generated file easier to check.

Should a DCL file name be specified that does not exist DCL code is placed in the generated DECW$PRIVATE_SERVER_SETUP.COM to display a message on the console indicating the problem, as well as a comment line message indicating the non-included file name. If any errors of this nature occured DCL code is also generated to display another warning message on the console at the conclusion of server private startup.

Of course, the CONFGR utility does not assess the correctness of the DCL specified in the requirement files!

Do not directly edit any VAXcluster member's SYS$SPECIFIC:[SYSMGR]DECW$PRIVATE_SERVER_SETUP.COM file, use the CONFGR utility.

Example Generated DECW$PRIVATE_SERVER_SETUP.COM

```
$! DECW$PRIVATE_SERVER_SETUP.COM for WS1
$! Created: Friday, 15-MAY-1992 08:34
$! Produced by: $1$dua8:[site_manager]confgr.exe
$!
$! DO NOT MODIFY this file directly, use the CONFGR utility!
$!
$ set noon
$ say = "WRITE SYS$OUTPUT"
$!
$!------------------------- 100DPI ----------------------------------!
$ decw$server_density == 100
```

6.4 Using CONFGR

There are 4 basic steps when using the CONFGR utility.

1. Create or modify any requirement DCL files, as required.
2. Edit the CONFGR.DECW configuration file, as required.
3. Generate the system-specific DECW$PRIVATE_SERVER_SETUP.COM file.

   This will be created in the CONFGR$$DECWINDOWS directory and named DECW$PRIVATE_SERVER_SETUP.*system-name*.

   The CONFGR command for this activity is:
   ''CONFGR /DECWINDOWS [/NODE=*name*[,*name*]]''

   This has not altered any system's actual DECW$PRIVATE_SERVER_SETUP.COM. It is possible at this stage to review, change, delete, etc., any of the configuration files.
4. Distribute the generated DECW$PRIVATE_SERVER_SETUP.COM files to system-specific directories.

   This copies the DECW$PRIVATE_SERVER_SETUP.*system-name* file, from the CONFGR$$DECWINDOWS directory to the system's specific root, *disk*:[*sysroot*.SYSMGR]. It is renamed from DECW$PRIVATE_SERVER_SETUP.*system-name* to DECW$PRIVATE_SERVER_SETUP.COM.

   The CONFGR command for this activity is:
   ''CONFGR /DECWINDOWS /COPY [/NODE=*name*[,*name*]]''

6.4.1 Example sessions

Generating DECW$PRIVATE_SERVER_SETUP.COM Files

```
DAJAV$ confgr /decwindows /node=ws1,ws2,ws3,ws4
Configure DECW$PRIVATE_SERVER_SETUP.COM for WS1? [Y]:
%CONFGR-I-NOW, DECW$PRIVATE_SERVER_SETUP.COM for WS1
Configure DECW$PRIVATE_SERVER_SETUP.COM for WS2? [Y]:
%CONFGR-I-NOW, DECW$PRIVATE_SERVER_SETUP.COM for WS2
Configure DECW$PRIVATE_SERVER_SETUP.COM for WS3? [Y]:
%CONFGR-I-NOW, DECW$PRIVATE_SERVER_SETUP.COM for WS3
Configure DECW$PRIVATE_SERVER_SETUP.COM for WS4? [Y]: n
%CONFGR-I-CONFGRED, WS1,WS2,WS3
%CONFGR-W-NOTCONFGRED, WS4
DAJAV$
```

Distributing DECW$PRIVATE_SERVER_SETUP.COM Files

```
DAJAV$ confgr /decwindows /node=ws1,ws2,ws3,ws4 /copy
%CONFGR-I-NOW, checking system roots on $1$DUA4:
%CONFGR-I-NOW, checking system roots on $1$DUA5:
%CONFGR-I-NOW, checking system roots on $1$DUA6:
%CONFGR-I-NOW, checking system roots on $1$DUA7:
%CONFGR-I-NOW, checking system roots on $1$DUA8:
%CONFGR-I-NOW, checking system roots on $1$DUA9:
%CONFGR-I-NOW, checking system roots on $1$DUA10:
%CONFGR-I-NOW, checking system roots on $1$DUA11:
%CONFGR-I-NOW, checking system roots on $1$DUA16:
%CONFGR-I-NOW, checking system roots on $1$DUA17:
%CONFGR-I-NOW, checking system roots on $1$DUA22:
%CONFGR-I-NOW, checking system roots on $1$DUA23:
%CONFGR-I-NOW, checking system roots on $1$DUA24:
%CONFGR-I-NOW, checking system roots on $1$DUA25:
%CONFGR-I-NOW, checking system roots on $1$DUA34:
%CONFGR-I-NOW, checking system roots on $1$DUA35:
%CONFGR-I-NOW, checking system roots on $1$DUA36:
%CONFGR-I-NOW, checking system roots on $1$DUA37:
%CONFGR-I-NOW, checking system roots on $1$DUA0:
%CONFGR-I-NOW, checking system roots on $1$DUA1:
%CONFGR-I-NOW, checking system roots on $1$DUA2:
%CONFGR-I-NOW, checking system roots on $1$DUA12:
%CONFGR-I-NOW, checking system roots on $1$DUA13:
%CONFGR-I-NOW, checking system roots on $1$DUA14:
%CONFGR-I-NOW, checking system roots on $1$DUA15:
%CONFGR-I-NOW, checking system roots on $1$DUA18:
%CONFGR-I-NOW, checking system roots on $1$DUA19:
%CONFGR-I-NOW, checking system roots on $1$DUA20:
%CONFGR-I-NOW, checking system roots on $1$DUA21:
%CONFGR-I-NOW, checking system roots on $1$DUA30:
%CONFGR-I-NOW, checking system roots on $1$DUA31:
%CONFGR-I-NOW, checking system roots on $1$DUA32:
%CONFGR-I-NOW, checking system roots on $1$DUA33:
%CONFGR-I-NOW, checking system roots on SERV2$DUA0:
%CONFGR-I-NOW, checking system roots on SERV2$DUA1:
%CONFGR-I-NOW, checking system roots on SERV2$DUA2:
%CONFGR-I-NOW, checking system roots on SERV2$DUA3:
%CONFGR-I-NOW, checking system roots on SERV2$DUB255:
%CONFGR-I-NOW, checking system roots on DBOOT$DUA0:
%CONFGR-I-NOW, checking system roots on DBOOT$DUA4:
Copy DECW$PRIVATE_SERVER_SETUP.COM for WS3 to DBOOT$DUA4:[SYS13.SYSMGR]? [Y]: n
```

```
%CONFGR-I-NOW, checking system roots on DBOOT$DUA5:
Copy DECW$PRIVATE_SERVER_SETUP.COM for WS1 to DBOOT$DUA5:[SYS10.SYSMGR]? [Y]: n
Copy DECW$PRIVATE_SERVER_SETUP.COM for WS2 to DBOOT$DUA5:[SYS18.SYSMGR]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$DECWINDOWS]DECW$PRIVATE_SERVER_SETUP.WS
%PURGE-I-NOFILPURG, no files purged
%CONFGR-I-NOW, checking system roots on DBOOT$DUA6:
Copy DECW$PRIVATE_SERVER_SETUP.COM for WS4 to DBOOT$DUA6:[SYS15.SYSMGR]? [Y]:
%COPY-S-COPIED, SITE$MANAGEMENT:[SITE_MANAGER.CONFGR$$DECWINDOWS]DECW$PRIVATE_SERVER_SETUP.WS
%PURGE-I-NOFILPURG, no files purged
%CONFGR-I-NOW, checking system roots on DBOOT$DUB255:
%CONFGR-I-CONFGRED, WS2,WS4
%CONFGR-W-NOTCONFGRED, WS1,WS3
DAJAV$
```

Chapter 7
CONFGR Utility

The CONFGR utility is a command-line driven utility with a *DCL- like* interface. Qualifiers control its activity. Defaults attempt to specify the most useful or safe behaviours.

Installation:

CONFGR == "$location:CONFGR"

Usage:

CONFGR [*/qualifiers*]

CONFGR$$STARTUP:

This is the startup configuration directory. It contains:

- Startup DCL files/procedures
  These must be named *activity*.COM
- Configration file, CONFGR.DAT
- Generated, system-specific SYSTARTUP_V5.*system-name*

CONFGR$$SHUTDWN:

This is the shutdown configuration directory. It contains:

- Shutdown DCL files/procedures
  These must be named *activity*.COM
- Configration file, CONFGR.DAT
- Generated, system-specific SYSHUTDWN.*system-name*

CONFGR$$MODPARAMS:

This is the Modparams/AUTOGEN configuration directory. It contains:

- SYSGEN and AUTOGEN-directive parameter files
  These must be named *purpose*.PAR
- Configration file, CONFGR.PAR
- Generated, system-specific MODPARAMS.*system-name*

CONFGR$$DECWINDOWS:

This is the DECwindows configuration directory. It contains:

- DCL files/procedures
  These must be named *requirement*.COM
- Configration file, CONFGR.DECW
- Generated, system-specific DECW$PRIVATE_SERVER_SETUP.*system-name*

7.1 qualifiers

/CONFIRM (default)
/NOCONFIRM
CONFGR will ask for confirmation before any activity involving file changes.

/COPY
Copy the generated file to the system's root directory. CONFGR searches all disks accessable from that node looking for [SYS0] . . . [SYS*n*] directories. When located the [.SYSEXE]MODPARAMS.DAT is examined to determine the system name from the SCSNODE parameter.
By default CONFGR generates files, not distribute them.

/DECWINDOWS
Generate DECW$PRIVATE_SERVER_SETUP.COM files.
By default Startup, Shutdown, Modparams/AUTOGEN and DECwindows files are configured. However if this qualifier is used it causes the other not to be generated.

/LOCASE (default)
/NOLOCASE
Convert non-quoted text to lower case before placing it in the generated file.

/MODPARAMS
Generate MODPARAMS.DAT files.
By default Startup, Shutdown, Modparams/AUTOGEN and DECwindows files are configured. However if this qualifier is used it causes the other not to be generated.

/NODE=*name*[,*name*]
By default CONFGR operates on all systems in the configuration file. This qualifier allows the specification of one, or more systems, in a comma- separated list. Acitivty will be confined to the specified system(s).

/PURGE
Purge the specified configuration directory to remove old versions of files.

/SHUTDWN
Generate SYSHUTDWN.COM files.
By default Startup, Shutdown, Modparams/AUTOGEN and DECwindows files are configured. However if this qualifier is used it causes the other not to be generated.

/STARTUP
Generate SYSTARTUP_V5.COM files.
By default Startup, Shutdown, Modparams/AUTOGEN and DECwindows files are configured. However if this qualifier is used it causes the other not to be generated.

/UPCASE
/NOUPCASE (default)
Convert non-quoted text to upper case before placing it in the generated file.

7.2 examples

Generating SYSTARTUP_V5.COM for all systems in the configuration after making a change to a fundamental startup DCL file. We know what we're doing, turn off the default confirmation.

```
$ CONFGR /STARTUP /NOCONFIRM
```

Distributing the SYSTARTUP_V5.COM for all systems in the configuration.

```
$ CONFGR /STARTUP /NOCONFIRM /COPY
```

Generating MODPARAMS for all systems in the configuration after making a change to a fundamental parameter file.

```
$ CONFGR /MODPARAMS /NOCONFIRM
```

Distributing the MODPARAMS.DAT for all systems in the configuration.

```
$ CONFGR /MODPARAMS /NOCONFIRM /COPY
```

Having introduced a new system into the configuration, and edited the CONFGR$$STARTUP:CONFGR.DAT for startup and the CONFGR$$STARTUP:CONFGR.PAR for AUTOGEN, now get CONFGR to generate both at once.

```
$ CONFGR /NOCONFIRM /NODE=system-name
```

Index

C

CONFGR utility, 7–1
  directories
    CONFGR$$DECWINDOWS:, 7–2
    CONFGR$$MODPARAMS:, 7–2
    CONFGR$$SHUTDWN:, 7–1
    CONFGR$$STARTUP:, 7–1
  qualifiers, 7–2
  usage examples, 7–3

D

DECwindows, 6–1
  basic steps, 6–5
  configuration control, 6–1
  configuration file, 6–2
    CONFGR.DECW, 6–2
    example, 6–2
  DCL files
    example, 6–2
  do not edit DECW$PRIVATE_SERVER_
    SETUP.COM, 6–1
  generated DECW$PRIVATE_SERVER_
    SETUP.COM, 6–4
    do not edit, 6–4
    example, 6–5
  generation and distribution, 6–1
  requirement DCL files, 6–2
  using CONFGR
    example sessions, 6–5

M

Modparams/AUTOGEN, 5–1
  basic steps, 5–8
  check AGEN$PARAMS.REPORT, 5–2
  configuration control, 5–1
  configuration file, 5–3
    CONFGR.PAR, 5–3
Modparams/AUTOGEN
  configuration file (cont'd)
    example, 5–3
  do not edit MODPARAMS.DAT, 5–2
  example sessions, 5–8
  generated MODPARAMS.DAT, 5–6
    do not edit, 5–7
    example, 5–7
  generation and distribution, 5–1
  no consistency checking, 5–2
  parameter files, 5–2
    example 1, 5–2
    example 2, 5–3

S

Shutdown, 4–1
  basic steps, 4–3
  configuration control, 4–1
  configuration file, 4–2
    CONFGR.DAT, 4–2
    sequencing activities, 4–2
  do not edit SYSHUTDWN.COM, 4–1
  generated SYSHUTDWN.COM, 4–2
    do not edit, 4–2
  generation and distribution, 4–1
  sequencing activities, 4–2
  shutdown DCL files, 4–2
    must execute stand-alone, 4–2
Startup, 3–1
  basic steps, 3–7
  configuration control, 3–1
  configuration file, 3–3
    CONFGR.DAT, 3–3
    example, 3–4
    sequencing activities, 3–4
  do not edit SYSTARTUP_V5.COM, 3–2
  generated SYSTARTUP_V5.COM, 3–4
    do not edit, 3–5

Startup
generated SYSTARTUP_V5.COM (cont'd)
efficient, inline code, 3–4
example, 3–5
generation and distribution, 3–1
sequencing activities, 3–4
startup DCL files, 3–2
example 1, 3–2
example 2, 3–2
must execute stand-alone, 3–3
using CONFGR
example sessions, 3–7