

OpenVMS STARTUP – Underappreciated Flexibility

Robert Gezelter

Robert Gezelter Software Consultant
Flushing, New York 11358-1731 USA
gezelter@rlgsc.com
<http://www.rlgsc.com>

The OpenVMS STARTUP process is documented, but the information is spread across several different manuals and files. This leads to a fragmentation of knowledge and the propagation of out-of-date and apochryphal information. A close reading of the documentation and associated files yields a coherent picture of the STARTUP facility and how it can be used to improve time-to-restart.

Keywords: OpenVMS · STARTUP · localization.

1 Introduction

OpenVMS systems are highly reliable. Restarts are infrequent. It is not unheard of for an OpenVMS system to go years between restarts. I have encountered individuals whose multi-year job tenure was marked by not a single restart.

However, infrequent is not never. Every system starts, and eventually, every system shuts down. Shutdowns can be controlled or uncontrolled. Individual system components will also need to be quiesced and restarted at some point, if for no other reason than software updates.

The context of startup and shutdown is also changing with the advent of cloud-hosted OpenVMS instances. The lifespan of cloud-based instances can be quite short. Cloud instances are spun up (activated) or spun down (deactivated) based on workload or other factors. Startup and/or shutdown may also be fragmented, with a need to separate “system” elements from “application” elements.

For reasons relating to real-time response, cloud-based instances may be started and then placed in a suspended state, allowing later rapid activation. Some more application-oriented elements may not be amenable to suspension, and will need to be started as part of the subsequent activation process. The converse applies when suspending an instance. Some elements may need to be quiesced, even though the basic instance will not be shutdown, merely suspended.

Orderly shutdowns are desired. Uncontrolled shutdowns (e.g., hitting the halt switch; a crash) leave a trail of data detritus, from out-of-date storage bitmaps, to unflushed buffers and out-of-date or corrupted indices in keyed files and databases. Most of the damage can be audited out, but such corrective measures are time-consuming and delay return to service. Orderly shutdowns, which

ensure that databases and file systems are properly quiesced and disconnected, speed the inevitable restart and return to service.

Testing startups and shutdowns has become easier. In the not so distant past, it was necessary to schedule access to physical hardware for testing purposes. Now, VAX, Alpha, and x86-64, virtual machines exist. This allows testing of startup and shutdowns on virtual machines, residing on Windows laptops or desktops. Virtual machines are also easier to reset when a modification goes awry. [1]

2 Methodology

STARTUP.COM is documented in multiple places in the OpenVMS documentation set, including the *OpenVMS System Manager's Manual: Volume 1: Essentials* [2] and the *OpenVMS System Management Reference Manual*. [3] However, it should be noted that these manuals were first written before the advent of the Version 5.0 STARTUP.COM, and it is often not clear if the documentation writers had a deep understanding of the capabilities of STARTUP.COM.

An exegesis of STARTUP.COM yields additional information. However, as is the case with any other composition, software or literary, it is admittedly challenging to separate designer intent from accidents of composition.

The OpenVMS Engineering-supplied command and template localization exit command files have similar issues as evidence. Some were written with a clear understanding of what appears to be the intent of the database-driven STARTUP.COM. Some are vestiges of the pre-Version 5.0 STARTUP.COM. Yet others had things inserted that imply the individual doing the alternation did not understand the structure beyond the fact that SYSTARTUP_VMS.COM was executed during STARTUP.COM. I have tried to discount the aberrations and emphasize what appears to be the original intent. Except where otherwise noted, the files used for my exegesis were OpenVMS 8.4.

3 History

STARTUP.COM assumed its present form with VAX/VMS 5.0 in 1988. (The system was rebranded as OpenVMS in 1991 with the release of OpenVMS 5.4-2). The present STARTUP.COM uses a collection of RMS keyed files, referred to as databases, to identify tasks to be performed during STARTUP.

4 In the Beginning

Once the secondary bootstrap has completed its processing, control is transferred to SYSINIT via the entry point EXE\$INIT. [4, 5] SYSINIT then initializes the processor and memory management and builds permanent OpenVMS data structures, including the SYS\$SYSTEM and SYS\$COMMON logical names in the System

```

"ACP$BADBLOCK_MBX" = "MBA4:"
"DISK$AXPSYS084" = "ANYSYS$DKAO:"
"SYS$COMMON" = "ANYSYS$DKAO:[SYS0.SYSCOMMON.]"
"SYS$DISK" = "ANYSYS$DKAO:"
"SYS$LOADABLE_IMAGES" = "SYS$SYSROOT:[SYS$LDR]"
"SYS$MESSAGE" = "SYS$SYSROOT:[SYSMSG]"
"SYS$P$SHARED_REGISTER" = ".K....."
"SYS$SHARE" = "SYS$SYSROOT:[SYSLIB]"
"SYS$SYSDEVICE" = "ANYSYS$DKAO:"
"SYS$SYSROOT" = "ANYSYS$DKAO:[SYS0.]"
                = "SYS$COMMON:"
"SYS$SYSTEM" = "SYS$SYSROOT:[SYSEXE]"
"SYS$TIMEZONE_DIFFERENTIAL" = "-14400"
"SYS$TOPSYS" = "SYS0"

```

Figure 1. LNM\$SYSTEM.TABLE contents at entry to STARTUP.COM

logical name table LNM\$SYSTEM.TABLE, Figure 1. It mounts the system disk and creates the STARTUP process. The default is set to <physical-device-name>:[SYSEXE].

The initialization code creates the STARTUP process. The STARTUP process then executes the STARTUP file, by default SYS\$SYSTEM:STARTUP.COM.

5 The Phases of STARTUP

STARTUP\$STARTUP_PHASES, which defaults to SYS\$STARTUP:VMS\$PHASES.DAT, enumerates the phases of STARTUP.COM.

As delivered in the OpenVMS distribution, VMS\$PHASES.DAT lists nine phases:

```

INITIAL
DEVICES
PRECONFIG
CONFIG
BASEENVIRON
LPBEGIN
LPMAIN
LPBETA
END

```

Of these phases, the OpenVMS Engineering-supplied STARTUP database file SYS\$SYSDEVICE:[SYS\$STARTUP]VMS\$VMS.DAT contains entries for all but the LPMAIN, and LPBETA phases. END is not an end-of-file sentinel, but the name of the last phase in the OpenVMS Engineering-supplied list of phases. The second of the two OpenVMS Engineering-supplied STARTUP database files, SYS\$SYSDEVICE:[SYS\$STARTUP]VMS\$LAYERED.DAT, is distributed empty.

These names are just that, identifiers. To be precise, a “Phase” is a 12-character unique identifier. They could have been named after the Greek alphabet (ALPHA,

BETA, GAMMA, DELTA, . . .); the International Civil Aviation Organization phonetic alphabet (ALFA, BETA, CHARLIE, DELTA, ECHO, FOXTROT, . . .); or the World Meteorological Organization annual naming sequence for named storms. The source or collating sequence of the phases does not matter. Each phase is processed in the sequence that it appears in the STARTUP phase file. All of the entries associated with a particular phase are processed before STARTUP.COM moves on to the next phase.

6 The STARTUP Databases

As shipped, there are two database files associated with STARTUP.COM:

- VMS\$VMS.DAT and
- VMS\$LAYERED.DAT,

both in SYS\$COMMON:[SYS\$STARTUP]. However, they are enumerated by STARTUP.COM using the Executive-mode system logical names STARTUP\$STARTUP_LIST and SYS\$STARTUP, which are defined by default at the beginning of STARTUP.COM, if not already defined externally before entry.

The database files are simple keyed RMS files, with the primary key being the phase name. The secondary key is the filename, with no duplicate key values allowed.

6.1 Execution Mode

Entries in the STARTUP databases can be set to execute in one of three ways:

- DIRECT The specified file is executed directly as part of the STARTUP process.
- SPAWN The specified file is executed as a spawned subprocess of the STARTUP process.
- BATCH The specified DCL command file is submitted under the SYSTEM account for execution in the default queue as a batch job.

As distributed, there is a single entry in VMS\$VMS.DAT show an execution mode of “CALLED”. Internally to STARTUP.COM, “CALLED” is converted to “DIRECT”.

There are clearly tradeoffs between the different execution modes. BATCH executions can be delayed and executed during later phases. SPAWN is more expensive in resources than DIRECT, but SPAWN provides the potential for parallel processing of longer, non-interfering tasks. Errors in command files processed using DIRECT can cause problems; procedures executed in subprocesses provide a degree of isolation and damage limitation.

If multiple files are to be executed using the SPAWN mode, care must be taken to ensure that the quotas of the STARTUP process and overall system resources are sufficient to support the multiple subprocesses. Overall, it is the author’s opinion that there should be a slight preference for SPAWN.

7 STARTUP Appendages

The VAX/VMS Version 5.0 re-imagining of STARTUP.COM divided the functionality of STARTUP.COM into a main file STARTUP.COM and a series of subsidiary files, with the STARTUP database files identified in Executive-mode system logical name STARTUP\$STARTUP_LIST.

STARTUP.COM is in essence a sequencer, with some special case code for manual startups of NETWORK and AUDIT_SERVER.

The significant work of system startup is done by what this article will refer to as *appendages*,¹ command files identified for sequenced execution in the STARTUP databases. These command files are referenced using the SYS\$STARTUP logical name.

Phase	Appendage (in SYS\$STARTUP)
INITIAL	VMS\$INITIAL-050_VMS.COM
	VMS\$INITIAL-050_LIB.COM
	VMS\$INITIAL-050_CONFIGURE.COM
	VMS\$CONFIG-050_CSP.COM
INITIAL	VMS\$CONFIG-050_SHADOW_SERVER.COM
DEVICES	VMS\$DEVICE_STARTUP.COM
PRECONFIG	IPC\$STARTUP.COM
CONFIG	VMS\$CONFIG-050_VMS.COM
	VMS\$CONFIG-050_ERRFMT.COM
	VMS\$CONFIG-050_CACHE_SERVER.COM
	VMS\$CONFIG-050_OPCOM.COM
	VMS\$CONFIG-050_AUDIT_SERVER.COM
	VMS\$CONFIG-050_JOBCTL.COM
	VMS\$CONFIG-050_LMF.COM
	VMS\$CONFIG-050_SECURITY_SERVER.COM
	VMS\$CONFIG-050_ACME_SERVER.COM
	BASEENVIRON
	VMS\$BASEENVIRON-050_SMISERVER.COM
	VMS\$BASEENVIRON-050_LIB.COM
	DECDTM\$STARTUP.COM
	LICENSE_CHECK.EXE
	VMS\$BASEENVIRON-050_INDICT_SERVER.COM
LPBEGIN	VMS\$LPBEGIN-050_STARTUP.COM
	VMS\$LPBEGIN-050_SMHANDLER.COM
END	VMS\$END-050_COORDINATED.COM

Table 1. OpenVMS Engineering-supplied Appendages. OpenVMS Version 8.4 as distributed

¹ The *OpenVMS System Manager's Manual* [2] refer to such files as *components*, however, the documentation also uses the term *components* for other meanings.

In most cases, these appendages are DCL command files which are invoked by `STARTUP.COM`, Table 1. There is also provision for execution of images (`.EXE` files) using the `DIRECT` mode. Appendages and routines invoked directly or indirectly from appendages have access to `LNM$SYSTEM.TABLE`, `LNM$STARTUP.TABLE`, and other system-wide logical name tables. Information can be passed from one appendage to one or more later executed appendages by this route, as is done by several appendages and `VMS$END-050.COORDINATED.COM`, which is executed during the `END` phase, as specified in the OpenVMS Engineering-supplied `VMS$VMS.DAT` database file.

8 STARTUP User-Modifiable Exits

The classic user-localization files (e.g., `SYSTARTUP.VMS.COM`, `SYPAGSWPFILES.COM`, and `SYLOGICALS.COM`) exist as “localization exits” invoked by various appendages at different points in this database-driven structure, Table 2. Localization exits

Phase	Appendage (in <code>SYS\$STARTUP</code>)	Filename (in <code>SYS\$MANAGER</code>)
DEVICE	<code>VMS\$DEVICE_STARTUP.COM</code>	<code>SYCONFIG.COM</code> <code>SYLOGICALS.COM</code> <code>SYPAGSWPFILES.COM</code>
CONFIG	<code>VMS\$CONFIG-050_AUDIT_SERVER.COM</code>	<code>SYSECURITY.COM</code>
BASEENVIRON	<code>VMS\$BASEENVIRON-050_INDICT_SERVER.COM</code>	<code>SYS\$INDICTMENT_POLICY.COM</code>
LPBEGIN	<code>VMS\$LPBEGIN-050_STARTUP.COM</code>	<code>SYSTARTUP.VMS.COM</code>

Table 2. OpenVMS Engineering-supplied Localization Exits, OpenVMS 8.4 as distributed

are guarded by an invocation of `F$SEARCH` for the appropriate filename to determine presence, followed by an invocation of the file if the file is found to be present. Disabling a localization file is therefore merely a matter of renaming the file in question.

9 OpenVMS Engineering-supplied Appendages

The OpenVMS Engineering-supplied appendages fall into seven of the nine OpenVMS Engineering-supplied phases. While phases `LPMAIN`, and `LPBETA` are defined in the distributed `VMS$PHASES.DAT`, there are not presently any activities associated with these phases as distributed.

All of the OpenVMS Engineering-supplied appendages execute serially in `DIRECT` mode. It is not absolutely clear if this serialization is necessary in all cases. As an example, many of the OpenVMS Engineering-supplied appendages executed during phase `CONFIG` do not appear to have mutual dependencies, which

would make SPAWN mode a possibility. On the other hand, there is little to be gained by parallelizing execution of a single RUN command.

9.1 Phase INITIAL

- VMS\$INITIAL-050_VMS.COM
Define majority of SYS\$* system logical names. On processors (Alpha, IA64) which support translated images, define logical names used to point to shared, translated images. If WRITESYSPARAMS was specified during a conversational bootstrap, use SYSGEN to update the permanent copy of the system parameter file.
- VMS\$INITIAL-050_LIB.COM
Define DBG\$* system logical names. Define TPU\$* system logical names. Define SYS\$LANGUAGE and related system logical names. Define RMS semantic tags. Define Time Zone formats and related system logical names.
- VMS\$INITIAL-050_CONFIGURE.COM
Start CONFIGURE process. If needed.
- VMS\$CONFIG-050_CSP.COM
If member of cluster, start CLUSTER_SERVER.
- VMS\$CONFIG-050_SHADOW_SERVER.COM
If shadowing present, start SHADOW_SERVER.

9.2 Phase DEVICES

- VMS\$DEVICE_STARTUP.COM
 - Configure devices.
 - If present, run SYS\$MANAGER:SYCONFIG.COM
 - If present, run SYS\$MANAGER:SYLOGICALS.COM
 - If present, run SYS\$MANAGER:SYPAGSWPFILES.COM
 - Run SYS\$STARTUP:USB\$STARTUP.COM
 - Run SYS\$STARTUP:LAN\$STARTUP.COM
 - Run SYS\$STARTUP:CLUE\$STARTUP.COM
 - Run SYS\$SYSTEM:FASTPATH_SERVER.EXE

9.3 Phase PRECONFIG

- IPC\$STARTUP.COM

9.4 Phase CONFIG

- VMS\$CONFIG-050_VMS.COM
Check page file. Set OPA0: characteristics (NOMODEM, NODISCONNECT, NODIALUP).
- VMS\$CONFIG-050_ERRFMT.COM
Start ERRFMT process.
- VMS\$CONFIG-050_CACHE_SERVER.COM
Start CACHE_SERVER process.

- VMS\$CONFIG-050_OPCOM.COM
Start OPCOM process.
- VMS\$CONFIG-050_AUDIT_SERVER.COM
If present, run SYS\$MANAGER:SYSECURITY.COM. Start AUDIT_SERVER process.
- VMS\$CONFIG-050_JOBCTL.COM
Start JOB_CONTROL process.
- VMS\$CONFIG-050_LMF.COM
If SYS\$SPECIFIC: [SYSEXE]LMF\$SYSTEM.LDB exists, load VAX-VMS license.
- VMS\$CONFIG-050_SECURITY_SERVER.COM
Start Security Server.
- VMS\$CONFIG-050_ACME_SERVER.COM
Start ACME server.

9.5 Phase BASEENVIRON

- VMS\$BASEENVIRON-050_VMS.COM
 - If present, run SYS\$STARTUP:DTSS\$UTC_STARTUP.COM
 - If present, run SYS\$STARTUP:TDF\$UCT_STARTUP.COM
 - Run SYS\$NET_SERVICES.COM
 - If present, run SYS\$STARTUP:NET\$STARTUP.COM
 - Run SYS\$STARTUP:ICC\$STARTUP.COM
 - Define MOUNT locally to MOUNT/NOASSIST
 - Set default interactive logins to 64.
- VMS\$BASEENVIRON-050_SMISERVER.COM
Start SMI_SERVER.
- VMS\$BASEENVIRON-050_LIB.COM
Set default to SYS\$SYSTEM. Run SYS\$SYSTEM:SMGMAPTRM.EXE
- DECDTM\$STARTUP.COM
- LICENSE_CHECK.EXE
Check licenses.
- VMS\$BASEENVIRON-050_INDICT_SERVER.COM
 - If present, execute SYS\$MANAGER:SYS\$INDICTMENT_POLICY.COM
 - If system executive-mode logical name SYS\$INDICT_START equals "1" then start INDICTMENT_SERVER.

9.6 Phase LPBEGIN

- VMS\$LPBEGIN-050_STARTUP.COM
 - If system executive-mode logical name SYS\$AUDIT_SERVER_INIHIBIT translates as "", then initiate the AUDIT_SERVER.
 - If SYS\$STARTUP:SSL*\$STARTUP.COM exists, run the file. NOTE: If there are two variants matching the search string, only the first will be executed.
 - If present, run SYS\$STARTUP:ENCRYPT_START.COM
 - If present, run SYS\$MANAGER:SYSTARTUP_VMS.COM
 - Execute DECwindows parameter checks and startup, if needed.
 - Enable interactive logins using SET LOGON/INTERACTIVE NOTE: This is probably a bug, logins would be better enabled during the END phase.)
- VMS\$LPBEGIN-050_SMHANDLER.COM
Start SMHANDLER after loading associated device driver.

9.7 Phase END

– VMS\$END-050.COORDINATED.COM

Depending upon the contents of logical names, start various layered products:

- SYS\$STARTUP:ACME.STARTUP.COM
- SYS\$STARTUP:DCOM\$STARTUP.COM
- SYS\$STARTUP:NTA\$STARTUP.COM
- SYS\$STARTUP:NTA\$AUTHENTICATED_RPC.COM
- SYS\$STARTUP:NTA\$STARTUP_NT_ACME.COM
- SYS\$STARTUP:REG\$STARTUP.COM

10 Modifying STARTUP-Invoked Elements

One can write one's own appendages and localization exits for execution during the STARTUP process. Appropriate levels of caution are highly recommended.

Errors in modifications and insertions in the STARTUP sequence will lead to an unusable system. Most of these problems can be corrected by doing a conversational bootstrap with SET /STARTUP=OPA: to temporarily delay the execution of STARTUP.COM, allowing remediation of the problem. Always preserve unmodified copies of files when making changes to existing files. Done carefully, modifications are feasible.

Adding a user-supplied appendage to the STARTUP sequence requires putting the user-supplied appendage file into a directory in the SYS\$STARTUP search list (e.g., SYS\$SPECIFIC:[SYSMGR] and entering the name and parameters in one of the STARTUP database files, e.g. VMS\$LAYERED.DAT.

User-supplied appendages can be tested, with appropriate care, outside of the STARTUP.COM context, easing debugging and testing.

11 Improvements

STARTUP.COM is not without its faults and limitations, some of which are conceptual. Solving the conceptual issues would require a significant effort. However, we should not let the perfect be the enemy of the good. Several of the shortcomings are relatively minor, and could be corrected with a small effort. Six of these issues are:

- The requirement that STARTUP.COM be run from the SYSTEM account.
- The special case processing of the NETWORK and AUDIT_SERVER options
- The requirement that the STARTUP\$STARTUP_* logical names be Executive-mode and contained in LNM\$SYSTEM.TABLE.
- More generally, the sequencing of the directories in the SYS\$STARTUP search list places SYS\$SYSROOT:[SYS\$STARTUP] first in the sequence, followed by SYS\$MANAGER, Figure 2. Switching the precedence of these two entries would have two related benefits:

```

"SYS$STARTUP" = "SYS$SYSROOT:[SYS$STARTUP]" (LNM$SYSTEM_TABLE)
    = "SYS$MANAGER"
1 "SYS$MANAGER" = "SYS$SYSROOT:[SYSMGR]" (LNM$SYSTEM_TABLE)

"SYS$SYSROOT" = "ANYSYS$DKAO:[SYSO.]" (LNM$SYSTEM_TABLE)
    = "SYS$COMMON:"
1 "SYS$COMMON" = "ANYSYS$DKAO:[SYSO.SYSCOMMON.]" (LNM$SYSTEM_TABLE)

```

Figure 2. Translation of `SYS$STARTUP` after initial `STARTUP.COM`

- it would allow modified versions of distributed files to temporarily override the distributed versions by simply being placed in `SYS$SPECIFIC:[SYSMGR]` or `SYS$COMMON:[SYSMGR]`;
 - would allow localization files templates in `SYS$SYSROOT:[SYS$STARTUP]` to be to be modified; with the modified versions placed in `SYS$SPECIFIC:[SYSMGR]` or `SYS$COMMON:[SYSMGR]`.
- The enabling of interactive logins at the end of file `VMS$LPBEGIN-050.STARTUP.COM` executed during phase `LPBEGIN` is disadvantageous. Enabling logins would be better scheduled as part of phase `END`. At present, logins are enabled before any of the tasks scheduled during phases `LPMAIN` and `LPBETA` are performed. It would likely be optimal if the enabling of logins (and the starting of individual queues) were segregated in discrete files at distribution, allowing increased flexibility.

Correction of these shortcomings would turn `STARTUP.COM` into a general-purpose sequencing utility. This could then be used to implement application startups and system and application shutdowns. These changes are particularly useful in the context of cloud-based OpfsenVMS instances, where shutdowns and startups will likely be far more frequent than is the case with bare-iron instances. In this changed context, the ability to structure partial startups and shutdowns increases usability significantly.

Removal of the requirements of the `SYSTEM` account and the logical name restrictions (e.g., Executive-mode and System logical name table) would broaden the applicability of `STARTUP.COM/SYSMAN` for additional purposes. There would be no reduction in security, as affecting system operation would still require appropriate privileges. However, non-`SYSTEM` startups/shutdowns could be accomplished with the minimum necessary privilege (e.g., `GROUP`, `DETACHED`, `PRMMBX`, `GRPNAM`) and not require the use of system-compromising privileges (e.g., `SYSNAM`, `CMKRNL`).

12 Conclusion

`STARTUP.COM` is less well understood than it should be. Appropriate use of the facilities provided by `STARTUP.COM` can shorten the return-to-service time, as well as provide a better way to manage system shutdown processing.

References

1. R. Gezelter, “Disposable Virtual Machines: Deliberately Expendable,” in *Ruminations – An IT Blog*, (<http://www.rlgsc.com/blog/ruminations/disposable-virtual-machines.html>), Aug. 2010.
2. VSI, *OpenVMS System Manager’s Manual, Volume 1: Essentials*. 2019. DO-DSYMV1-01A.
3. VSI, *OpenVMS System Management Utilities Reference Manual, Volume II M–Z*. 2020. DO-DSYVR2-01A.
4. DEC, *VAX/VMS Internals and Data Structures*. Digital Equipment Corporation, 1981. AA-K785A-TE.
5. R. Goldenberg and S. Saravanan, *OpenVMS AXP Internals and Data Structures, Version 1.5*. Digital Press, 1994.