

Thursday, October 23, 2025, 11:00 AM, Gardner Room

Inheriting and Structuring Logical Names

2025 OpenVMS Bootcamp
Portsmouth Sheraton
Portsmouth, New Hampshire USA

Robert Gezelter Software Consultant

Agenda

References

Introduction

OpenVMS Logical Name Context

Roots of Inefficiency

. . .

Agenda –

- Inefficient use of logical names is expensive
- Inefficiency impedes performance
- Efficiency is not difficult
- Maintainability
- Defining Logical Names
- Logical Name Context
- “Texture”

Robert Gezelter Software Consultant

Agenda

References

Introduction

OpenVMS Logical Name Context

Roots of Inefficiency

References

- OpenVMS System Service Manuals
- OpenVMS Programming Concepts Manual
- “Inheritance-based Environments in Stand-Alone OpenVMS Systems and OpenVMScusters”
OpenVMS Technical Journal, Volume 3, February 2004

Robert Gezelter Software Consultant

Agenda

References

Introduction

OpenVMS Logical Name Context

Roots of Inefficiency

. . .

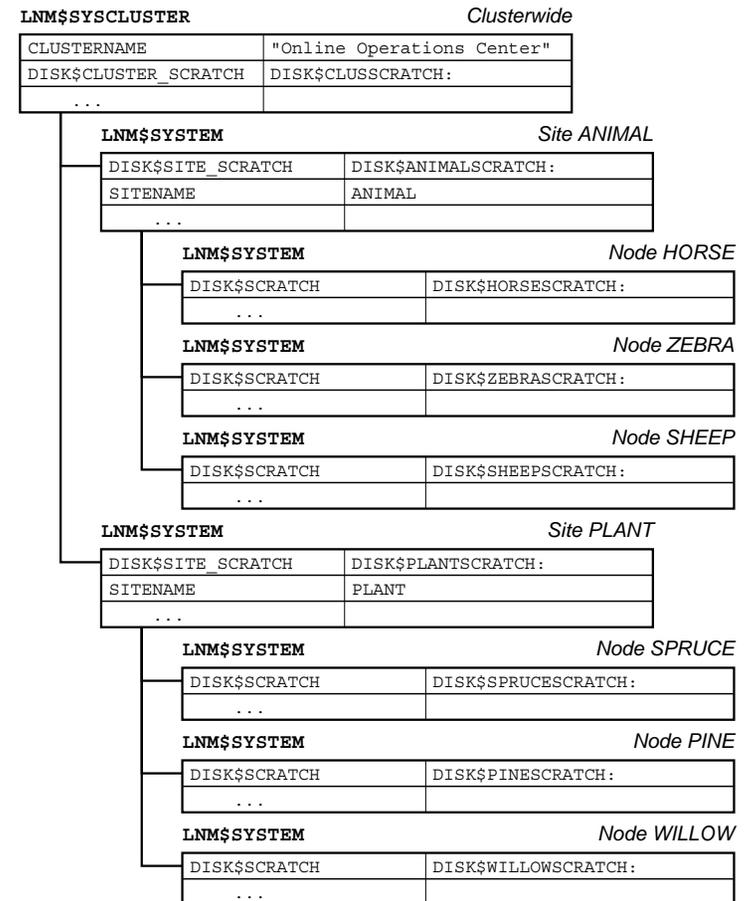
Introduction

Robert Gezelter Software Consultant

Agenda
References
Introduction
OpenVMS Logical Name Context
Roots of Inefficiency

OpenVMS Logical Name Context

- Hierarchical
- Hierarchy can be modified
- Higher levels provide defaults if not overridden by lower levels



Roots of Inefficiency – Unthinking Defaults

- SPAWN defaults to /LOGICAL_NAMES
- large LNM\$PROCESS
- Often a nested problem
- LNM\$_FILE_DEV is a partial solution
- LNM\$FILE_DEV default is in LNM\$SYSTEM_DIRECTORY
- LNM\$FILE_DEV can be overridden in LNM\$PROCESS_DIRECTORY

Robert Gezelter Software Consultant

OpenVMS Logical Name Context

Roots of Inefficiency

Demonstration

Note about LNM\$FILE_DEV

Summary

Define Logical Names

Local Results

Results – SPAWN/LOGICAL_NAMES

Results – SPAWN/NOLOGICAL_NAMES

A demonstration: Define LNM\$FILE_DEV and SCRATCH

- Define LNM\$FILE_DEV and SCRATCH
- No privileges necessary – other than PRCLM

```
$ assign/table=lnm$process_directory lnm$process,lnm$job,lnm$group,lnm$system -
$_ lnm$file_dev
$ show logical lnm$file_dev/table=lnm$process_directory
  "LNM$FILE_DEV" = "LNM$PROCESS" (LNM$PROCESS_DIRECTORY)
    = "LNM$JOB"
    = "LNM$GROUP"
    = "LNM$SYSTEM"
1  "LNM$PROCESS" = "LNM$PROCESS_TABLE" (LNM$PROCESS_DIRECTORY)
1  "LNM$JOB" = "LNM$JOB_81C79700" (LNM$PROCESS_DIRECTORY)
1  "LNM$GROUP" = "LNM$GROUP_000004" (LNM$PROCESS_DIRECTORY)

$ assign sys$sysdevice: scratch
$ sho log scratch
  "SCRATCH" = "SYS$SYSDEVICE:" (LNM$PROCESS_TABLE)
1  "SYS$SYSDEVICE" = "ODYSS2$DKA0:" (LNM$SYSTEM_TABLE)
```

Robert Gezelter Software Consultant

OpenVMS Logical Name Context

Roots of Inefficiency

Demonstration

Note about LNM\$FILE_DEV

Summary

Define Logical Names

Local Results

Results – SPAWN/LOGICAL_NAMES

Results – SPAWN/NOLOGICAL_NAMES

Local Results

- SCRATCH is in LNM\$PROCESS
- LNM\$FILE_DEV is in LNM\$PROCESS_DIRECTORY

```
$ show logical lnm$file_dev/table=lnm$directory

"LNMF$FILE_DEV" = "LNMF$PROCESS" (LNMF$PROCESS_DIRECTORY)
  = "LNMF$JOB"
  = "LNMF$GROUP"
  = "LNMF$SYSTEM"
1 "LNMF$PROCESS" = "LNMF$PROCESS_TABLE" (LNMF$PROCESS_DIRECTORY)
1 "LNMF$JOB" = "LNMF$JOB_81C79700" (LNMF$PROCESS_DIRECTORY)
1 "LNMF$GROUP" = "LNMF$GROUP_000004" (LNMF$PROCESS_DIRECTORY)
1 "LNMF$SYSTEM" = "LNMF$SYSTEM_TABLE" (LNMF$SYSTEM_DIRECTORY)
  = "LNMF$SYSCLUSTER"
2 "LNMF$SYSCLUSTER" = "LNMF$SYSCLUSTER_TABLE" (LNMF$SYSTEM_DIRECTORY)
"LNMF$FILE_DEV" = "LNMF$PROCESS" (LNMF$SYSTEM_DIRECTORY)
  = "LNMF$JOB"
  = "LNMF$GROUP"
  = "LNMF$SYSTEM"
  = "DECW$LOGICAL_NAMES"
1 "LNMF$PROCESS" = "LNMF$PROCESS_TABLE" (LNMF$PROCESS_DIRECTORY)
1 "LNMF$JOB" = "LNMF$JOB_81C79700" (LNMF$PROCESS_DIRECTORY)
1 "LNMF$GROUP" = "LNMF$GROUP_000004" (LNMF$PROCESS_DIRECTORY)
1 "LNMF$SYSTEM" = "LNMF$SYSTEM_TABLE" (LNMF$SYSTEM_DIRECTORY)
  = "LNMF$SYSCLUSTER"
2 "LNMF$SYSCLUSTER" = "LNMF$SYSCLUSTER_TABLE" (LNMF$SYSTEM_DIRECTORY)
1 "DECW$LOGICAL_NAMES" [table] = "" (LNMF$SYSTEM_DIRECTORY)
$ sho log scratch
"SCRATCH" = "SYS$SYSDEVICE:" (LNMF$PROCESS_TABLE)
1 "SYS$SYSDEVICE" = "ODYSS2$DKA0:" (LNMF$SYSTEM_TABLE)
```

Inheriting and Structuring Logical Names

Slide 8

© 2025, Robert Gezelter Software Consultant, All Rights Reserved

Robert Gezelter Software Consultant

OpenVMS Logical Name Context

Roots of Inefficiency

Demonstration

Note about LNM\$FILE_DEV

Summary

Define Logical Names

Local Results

Results – SPAWN/LOGICAL_NAMES

Results – SPAWN/NOLOGICAL_NAMES

Experiment – SPAWN/LOGICAL_NAMES

- \$ SPAWN/LOGICAL_NAMES

- Caution: Do NOT abbreviate – /LOG is also valid

- Note SCRATCH

- Note LNM\$FILE_DEV

```
$ spawn
%DCL-S-SPAWNED, process GEZELTER_13338 spawned
%DCL-S-ATTACHED, terminal now attached to process GEZELTER_1
$ show logical/process
```

```
(LNM$PROCESS_TABLE)
```

```
"SCRATCH" = "SYS$SYSDEVICE:"
"SYS$COMMAND" = "_ODYSS2$OPA0:"
"SYS$DISK" = "SYS$SYSDEVICE:"
"SYS$ERROR" = "_ODYSS2$OPA0:"
"SYS$INPUT" = "_ODYSS2$OPA0:"
"SYS$OUTPUT" [super] = "_ODYSS2$OPA0:"
"SYS$OUTPUT" [exec] = "_ODYSS2$OPA0:"
"TT" = "_OPA0:"
```

```
$ show logical/table=lnm$process_directory
```

```
(LNM$PROCESS_DIRECTORY)
```

```
"LNM$FILE_DEV" = "LNM$PROCESS"
= "LNM$JOB"
= "LNM$GROUP"
= "LNM$SYSTEM"
"LNM$GROUP" = "LNM$GROUP_000004"
"LNM$JOB" = "LNM$JOB_81C79700"
"LNM$PROCESS" = "LNM$PROCESS_TABLE"
"LNM$PROCESS_DIRECTORY" [table] = ""
```

Inheriting and Structuring Logical Names

Slide 9

© 2025, Robert Gezelter Software Consultant, All Rights Reserved

Robert Gezelter Software Consultant

OpenVMS Logical Name Context

Roots of Inefficiency

Demonstration

Note about LNM\$FILE_DEV

Summary

Define Logical Names

Local Results

Results – SPAWN/LOGICAL_NAMES

Results – SPAWN/NOLOGICAL_NAMES

Experiment – SPAWN/NOLOGICAL_NAMES

- \$ SPAWN/NOLOGICAL_NAMES

- Caution: Do NOT abbreviate – /NOLOG is also valid

- Faster, No copying of LNM\$PROCESS, etc.

- Note no SCRATCH

- Note no LNM\$FILE_DEV

```
$ spawn/nological_names
%DCL-S-SPAWNED, process GEZELTER_55617 spawned
%DCL-S-ATTACHED, terminal now attached to process GEZELTER_!
$ show logical scratch
%SHOW-S-NOTRAN, no translation for logical name SCRATCH
$ sho logical/table=lnm$process_directory
(LNM$PROCESS_DIRECTORY)
```

```
"LNM$GROUP" = "LNM$GROUP_000004"
"LNM$JOB" = "LNM$JOB_81C79700"
"LNM$PROCESS" = "LNM$PROCESS_TABLE"
"LNM$PROCESS_DIRECTORY" [table] = ""
"LNM$PROCESS_TABLE" [table] = ""
```

```
$ show logical /process
```

```
(LNM$PROCESS_TABLE)
```

```
"SYS$COMMAND" = "_ODYSS2$OPA0:"
"SYS$DISK" = "SYS$SYSDEVICE:"
"SYS$ERROR" = "_ODYSS2$OPA0:"
"SYS$INPUT" = "_ODYSS2$OPA0:"
"SYS$OUTPUT" [super] = "_ODYSS2$OPA0:"
"SYS$OUTPUT" [exec] = "_ODYSS2$OPA0:"
```

Inheriting and Structuring Logical Names

Slide 10

© 2025, Robert Gezelter Software Consultant, All Rights Reserved

Note about LNM\$FILE_DEV

- Default value is in table LNM\$SYSTEM_DIRERECTORY
- Logical name directories are listed in LNM\$DIRECTORIES
- OpenVMS default is LNM\$PROCESS, LNM\$JOB, LNM\$GROUP, LNM\$SYSTEM
- DECwindows appends DEC\$WLOGICAL_NAMES
- Other layered products also modify LNM\$FILE_DEV in either LNM\$SYSTEM_DIRECTORY or LNM\$PROCESS_DIRECTORY

Note about LNM\$FILE_DEV (continued)

- LNM\$SYSTEM_DIRECTORY contains TWO definitions for LNM\$FILE_DEV: EXECUTIVE and SUPERVISOR
- Be civil – insert your additions by inserting
- DO NOT just set to believed default

Summary

- Inheritance is significant for performance and correctness
- Be explicit and intentional about SPAWN/LOGICAL_NAMES and SPAWN/NOLOGICAL_NAMES: an account's LOGIN.COM can lead to subtle bugs
- When debugging using SPAWN, make sure to include SHOW LOGICAL/PROCESS and SHOW LOGICAL/TABLE=LNM\$PROCESS_DIRECTORY

Robert Gezelter Software Consultant

. . .
OpenVMS Logical Name Context
Roots of Inefficiency
Demonstration
Note about LNM\$FILE_DEV
Summary

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
<http://www.rlgsc.com>

Session Notes & Materials:

<http://www.rlgsc.com/openvms-bootcamp/2025/index.html>