VMS Software

SYS V5.0 ECO Kit for VSI OpenVMS IA-64 V8.4-2L1

Release Notes

Publication Date: October 2026

Operating System: VSI OpenVMS IA-64 V8.4-2L1

Kit Name: VMS842L1I_SYS-V0500

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

Table of Contents

1. Kit Name 3
2. Kit Description 3
    2.1. Installation Rating 3
    2.2. Reboot Requirement 3
    2.3. Version(s) of VSI OpenVMS to Which This Kit May Be Applied 3
3. Kits Superseded by This Kit 3
4. Kit Dependencies 3
5. Problems Addressed in This Kit 4
    5.1. Potential Process Crash During Exception Handling in Security Services 4
    5.2. Potential System Service SSRVEXCEPTION Crash Under Rare Conditions 4
    5.3. Incorrect Final Status in Accounting Record 5
6. Problems Addressed From Previous Kits 5
    6.1. Use of pthread pshared Objects May Cause INVEXCEPTN System Crash 5
    6.2. Inefficient Use of Cached KPBs 6
    6.3. Rare INVEXCEPTN Bugcheck Due to Access Violation in MMG$FIND_GSTE 7
    6.4. SSRVEXCEPT Bugcheck When Using Global Section Tracing 7
    6.5. $CRMPSC With Descending Address Range Maps Section Incorrectly 8
    6.6. Enable a Per-process P1 Pool History Ring Buffer 8
    6.7. Improve System Behavior During Transient Nonpaged Pool Exhaustion 9
    6.8. Rare Spurious INVEXCEPTN System Crash 10
    6.9. Error Log Entries for Non-fatal Bugchecks Contain No Useful Data 11
    6.10. System Crashes With DELCONPFN Bugcheck During Process Rundown 11
    6.11. $CRMPSC System Service Failure Due to Page Count Overflow 12
    6.12. SS$_PAGOWNVIO Error With Partially Mapped Global Section 12
    6.13. Spurious SECREFNEG Bugcheck During Global Section Processing 13
    6.14. Performance Degradation After Installing SYS V3.0 ECO Kit 13
    6.15. ANALYZE/CRASH Cannot Analyze System Crash Dump File 14
7. Images or Files Replaced 14
8. Installation Instructions 18
    8.1. Compressed File 18
    8.2. Installation Command 19
9. Copyright 19
10. Disclaimer of Warranty and Limitation of Liability 19
11. Patch ID 19

Appendix A. User-Selectable Control Options and Scripting Considerations 20

    A.1. Controlling Kit Behavior for Introductory Questions 20
    A.2. Standard Behavior for YES/NO Questions Asked During Kit Installation 21
    A.3. Installing a Kit From a Batch Job 21
    A.4. Deprecated Logical Names From HPE ECO Kits 22

1. Kit Name

VMS842L1I_SYS-V0500

2. Kit Description

2.1. Installation Rating

INSTALL_1: To be installed by all customers.

This installation rating serves as a guide to which customers should apply this remedial kit.

Reference the Disclaimer of Warranty and Limitation of Liability Statement.

2.2. Reboot Requirement

A reboot is required after installing this kit.

VMS Software, Inc. strongly recommends that a reboot be performed immediately after kit installation to avoid system instability.

If you are installing this kit in an OpenVMS cluster, any systems that share the same system disk with the installing system must also be rebooted in order to make use of the new images.

2.3. Version(s) of VSI OpenVMS to Which This Kit May Be Applied

VSI OpenVMS IA-64 V8.4-2L1

3. Kits Superseded by This Kit

VMS842L1I_PRCMGT-V0100
VMS842L1I_SYS-V0200
VMS842L1I_SYS-V0300
VMS842L1I_SYS-V0400

Note

There was no SYS V1.0 ECO kit for VSI OpenVMS IA-64 V8.4-2L1.

This kit is named V5.0 to allow a common patch version level across all current VSI OpenVMS versions which have equivalent functionality and defect repair.

---

4. Kit Dependencies

VMS842L1I_UPDATE-V0100

All new patch kits for VSI OpenVMS IA-64 V8.4-2L1 require the VMS842L1I_UPDATE-V0100 kit.

5. Problems Addressed in This Kit

5.1. Potential Process Crash During Exception Handling in Security Services

Problem Description

Exception handling within the security sub-system may trigger an unexpected process crash under certain conditions.

This problem is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SECURITY.EXE
[SYS$LDR]SECURITY.STB
[SYS$LDR]SECURITY_MON.EXE
[SYS$LDR]SECURITY_MON.STB

VSI Case Identifier

Jira BO-1871

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

5.2. Potential System Service SSRVEXCEPTION Crash Under Rare Conditions

Problem Description

Under unpredictable but very rare circumstances, a NaT Consumption Fault may occur and could result in a system crash.

This behavior only occurs on VSI OpenVMS IA-64 systems and is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SYSTEM_PRIMITIVES.EXE
[SYS$LDR]SYSTEM_PRIMITIVES.STB
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.STB

VSI Case Identifiers

Jiras BO-2072, BO-2142
Netsuite NS8283

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

5.3. Incorrect Final Status in Accounting Record

Problem Description

Under certain conditions, when a process aborts unexpectedly, the process termination accounting record may contain an incorrect final status code.

This behavior only occurs on VSI OpenVMS IA-64 systems and is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]EXCEPTION.EXE
[SYS$LDR]EXCEPTION.STB
[SYS$LDR]EXCEPTION_MON.EXE
[SYS$LDR]EXCEPTION_MON.STB

VSI Case Identifiers

Jiras BO-2042, BO-2043, BO-2177
Netsuites NS7912, NS8558

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

6. Problems Addressed From Previous Kits

6.1. Use of pthread pshared Objects May Cause INVEXCEPTN System Crash

Problem Description

A problem within `pthread pshared` object handling may cause a system crash with a footprint similar to the following:

```
Crash Time:        17-APR-2018 09:32:54.82
Bugcheck Type:     INVEXCEPTN, Exception while above ASTDEL
Node:              <nodename>
CPU Type:          HP rx6600  (1.59GHz/12.0MB)
VMS Version:       V8.4-2L1
Current Process:   JDBC_P4_00004DA
Current Image:     $1$DGA159:[SYS0.SYSCOMMON.RDB$JDBC.0703-5V0H8E]RDBJDBCEXEC73.EXE
Failing PC:        FFFFFFFF.805D13F0    EXE$PSHARED_MUTEX_UNBLOCK_C+003D0
Failing PS:        00000000.00000603
Module:            PROCESS_MANAGEMENT   (Link Date/Time: 27-JUL-2016 10:30:20.13)
Offset:            001028F0
```

The crucial information identifying this problem is the bugcheck type, INVEXCEPTN, and the failing PC, at some offset within a routine called EXE$PSHARED* such as EXE$PSHARED_MUTEX_UNBLOCK.

The crash is caused by a timing window which accidentally allows use of a recently deleted `pshared mutex` or condition value. Oracle JDBC is a known consumer of these object types and may experience the crash under certain loads. Other applications which use `pshared` objects have the same risk of a system crash.

This problem is corrected with this ECO kit.

Some additional sanity checking of `pshared` object types has also been added to prevent incorrect application calls which operate on the wrong object type.

Images and/or Files Affected

[SYS$LDR]PROCESS_MANAGEMENT.EXE
[SYS$LDR]PROCESS_MANAGEMENT.STB
[SYS$LDR]PROCESS_MANAGEMENT_MON.EXE
[SYS$LDR]PROCESS_MANAGEMENT_MON.STB

Quix and/or Bugzilla Cases Reporting This Problem

VSI Bugzilla 1323

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.2. Inefficient Use of Cached KPBs

Problem Description

The OpenVMS system maintains a cache of unused KPBs (kernel process blocks) to avoid extra overhead when these structures are allocated. These structures are used within the executive and various device drivers, and reside in S0/S1 system virtual address space.

Due to a problem in checking the cache availability, some requests for KPBs would allocate new structures when cached unused KPBs were available. This could result in wasted S0/S1 space and wasted CPU overhead to allocate the new structures.

This problem is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SYSTEM_PRIMITIVES.EXE
[SYS$LDR]SYSTEM_PRIMITIVES.STB
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.STB

Quix and/or Bugzilla Cases Reporting This Problem

Internal Bugzilla 1414
QXCM1001406945
PTR 75-126-444

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.3. Rare INVEXCEPTN Bugcheck Due to Access Violation in MMG$FIND_GSTE

Problem Description

Under rare circumstances, the system may crash with an INVEXCEPTN bugcheck during RMS global buffer rundown. An incorrect register usage could overwrite a GSD (Global Section Descriptor) address, which would cause an access violation in MMG$FIND_GSTE.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE
[SYS$LDR]SYS$VM.STB

Quix and/or Bugzilla Cases Reporting This Problem

QXCM1001477127

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.4. SSRVEXCEPT Bugcheck When Using Global Section Tracing

Problem Description

Global section tracing for problem identification and analysis may be enabled by setting bit 7 in MMG_CTLFLAGS. This tracing should only be used for debugging purposes and is not ever enabled by default.

Should a failure to create a global section occur while the tracing is enabled, the system may crash with a SSRVEXCEPT bugcheck. This was due to an incorrectly initialized field if the failure path was taken.

This problem is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE
[SYS$LDR]SYS$VM.STB

Quix and/or Bugzilla Cases Reporting This Problem

QXCM1001606217

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.5. $CRMPSC With Descending Address Range Maps Section Incorrectly

Problem Description

In 2004, support was added for 64-bit value ranges in $CRMPSC. Due to an accidental re-insertion of the 32-bit equivalent instructions, $CRMPSC calls with descending address ranges would not map the address space correctly. This bug has been latent in all OpenVMS versions since that time.

As this problem was found during code inspection for X86 porting activity and never reported by a customer, it appears that descending address range use is essentially non-existent.

This problem is corrected with this ECO kit should any new application code ever use descending address ranges for $CRMPSC.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE
[SYS$LDR]SYS$VM.STB

Quix and/or Bugzilla Cases Reporting This Problem

None.

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.6. Enable a Per-process P1 Pool History Ring Buffer

Problem Description

OpenVMS has a per-process P1 pool ring buffer that can be used for tracking down P1 pool memory leaks. This is enabled when the SYSTEM_CHECK system parameter is set to 1. Packets which are allocated but not deallocated are moved to the top of the ring buffer to aid in analysis. The SYSTEM_CHECK debugging mechanism enables debug, trace, and sanity checks in many portions of the operating system, with a correspondingly higher overhead for normal system operation.

Enhancements have been made to allow a dynamic P1 pool history ring buffer without the extra overhead of SYSTEM_CHECK=1. This is enabled by setting bit 7 in the ACTIVE system parameter POOLCHECK. Processes created after the bit is set will have the pool history buffer mechanism in place.

For example, to enable the P1 pool history buffers, assuming that POOLCHECK is currently zero:

```
$ MCR SYSGEN
SYSGEN>  USE ACTIVE
SYSGEN>  SET POOLCHECK %X80
SYSGEN>  WRITE ACTIVE
SYSGEN>  EXIT
```

The ring buffer can be viewed with the SDA command:

```
SHOW PROCESS/POOL=P1/RING
```

The ring buffer can be disabled for future processes by clearing bit 7 in the ACTIVE system parameter POOLCHECK.

Images and/or Files Affected

[SYS$LDR]PROCESS_MANAGEMENT.EXE
[SYS$LDR]PROCESS_MANAGEMENT.STB
[SYS$LDR]PROCESS_MANAGEMENT_MON.EXE
[SYS$LDR]PROCESS_MANAGEMENT_MON.STB
[SYS$LDR]SYSTEM_PRIMITIVES.EXE
[SYS$LDR]SYSTEM_PRIMITIVES.STB
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.STB

Quix and/or Bugzilla Cases Reporting This Problem

QXCM1001428470

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.7. Improve System Behavior During Transient Nonpaged Pool Exhaustion

Problem Description

Transient nonpaged pool expansion which nears the point of pool exhaustion could trigger CPUSPINWAIT, CLUEXIT, or other bugchecks while reclaiming memory from the lookaside lists, or because the variable nonpaged pool freelist has become very fragmented.

To alleviate this condition, the upper 1/16th of the nonpaged pool memory area (NPAGEVIR) is now reserved just for the variable nonpaged pool list. This allows quicker recovery from the transient usage spikes without massive lookaside list reclamation activity. This change will only affect systems whose nonpaged pool expands near to the NPAGEVIR maximum value, and thus will have no noticeable effect on most systems. Generally, systems should use AUTOGEN to set a reasonable minimum value for NPAGEDYN for expected pool use, and a larger reserve of potentially used memory for NPAGEVIR. A general rule of thumb is to have at least 20 to 25% extra virtual pool space available.

In addition, new controls have been added to the POOLCHECK system parameter to aid in tracking down nonpaged pool corruption. If SYSTEM_CHECK is 0, and bit 6 is set in POOLCHECK (with some other bits in the low byte clear), then the second byte of POOLCHECK (bits 8-15) may be used to reflect a particular packet size. If this second byte is a value between 1 and 254, then only packets of that byte value multiplied by 64 will be included in the Nonpaged Pool History Ring-Buffer (thus corresponding to packets from 64 to 16256 bytes in size). This will reduce the "noise" of other packet activity in the ring buffer by including only packets of the size being investigated.

For example, if you are interested in packets of size 576 bytes, you could set POOLCHECK to:

%x00000040	Will not poison and check packets, but will trace all packet sizes in the ring buffer. Although bit 6 is set, the second byte is zero indicating that no size has been specified.

%x00000940	Will not poison and check packets, but will trace the allocation and deallocation of 576 byte packets (9, in the second byte, 9*64=576 byte packets to check). Not poisoning and checking packets reduces the overhead of having POOLCHECK enabled slightly.
%x61640943	Will both poison and check all packets, but will include only packets fitting in a 576 byte packet in the nonpaged pool history ring-buffer.
%x61640003	Will poison and check all packet sizes, and trace packets of all sizes since bit 6 of the lower byte is not set.
%x616400FF	Will poison and check all packet sizes, and trace packets of all sizes in the ring buffer since the lower byte has all bits set.

Images and/or Files Affected

[SYS$LDR]SYSTEM_PRIMITIVES.EXE
[SYS$LDR]SYSTEM_PRIMITIVES.STB
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.STB

Quix and/or Bugzilla Cases Reporting This Problem

QXCM1001380245
QXCM1001399095
PTR 75-74-320

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

Workaround

Try to address whatever triggered the nonpaged pool expansion and exhaustion so the event does not recur, or increase NPAGEDYN and NPAGEVIR to larger values.

6.8. Rare Spurious INVEXCEPTN System Crash

Problem Description

Under very unusual circumstances the system may crash with an INVEXCEPTN bugcheck. The crash dump footprint will be an access violation in routine EXE$RESTORE_HPF_REGISTERS.

This problem is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SYSTEM_PRIMITIVES.EXE
[SYS$LDR]SYSTEM_PRIMITIVES.STB
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.STB

Quix and/or Bugzilla Cases Reporting This Problem

VSI Bugzilla 2509

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.9. Error Log Entries for Non-fatal Bugchecks Contain No Useful Data

Problem Description

On IA-64 systems, the information provided in the system error log for non-fatal bugchecks contained random uninitialized data.

The error log content has been corrected to provide the accurate data for the PC, PSL, Frame Pointer, and Stack Pointers. The general registers display is now also correct for the registers that can be accessed. Any register which cannot be accessed is displayed as zeroes instead of a random value.

Images and/or Files Affected

[SYS$LDR]EXCEPTION.EXE
[SYS$LDR]EXCEPTION.STB
[SYS$LDR]EXCEPTION_MON.EXE
[SYS$LDR]EXCEPTION_MON.STB

Quix and/or Bugzilla Cases Reporting This Problem

VSI Bugzillas 2878, 3162
QXCM5336788780

Release Version of VSI OpenVMS That Will Contain This Change

VSI OpenVMS IA-64 V8.4-2L3.

6.10. System Crashes With DELCONPFN Bugcheck During Process Rundown

Problem Description

The system may crash with a DELCONPFN bugcheck while deleting the process page table during process rundown.

This problem only occurs on systems with more than 128 GB of memory-resident global sections while deleting the process mapping to such global sections.

This problem is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE
[SYS$LDR]SYS$VM.STB

VSI Case Identifier

QXCM1001722447

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

6.11. $CRMPSC System Service Failure Due to Page Count Overflow

Problem Description

The $CRMPSC system service may incorrectly handle an overflow of the page count parameter. This issue is most likely encountered when the relative page argument (RELPAG) is used.

This problem is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE
[SYS$LDR]SYS$VM.STB

VSI Case Identifier

Jira BO-653

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

6.12. SS$_PAGOWNVIO Error With Partially Mapped Global Section

Problem Description

The $CRMPSC system service may return an SS$_PAGOWNVIO error when encountering a range of demand zero pages to remap from previous calls. This is an obscure condition which most customers will never encounter.

This problem is corrected with this ECO kit.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE
[SYS$LDR]SYS$VM.STB

VSI Case Identifier

None, found during internal testing.

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

6.13. Spurious SECREFNEG Bugcheck During Global Section Processing

Problem Description

A synchronization issue when updating a structure that maintains the reference count for a global section could result in the reference count becoming incorrect. If this should happen, the reference count could become either too low or too high.

If the reference count becomes too low, this can lead to SECREFNEG system crashes. If the reference count becomes too high, then a global section may not get deleted when there are no longer any processes referencing the global section.

This patch corrects the synchronization issue when manipulating the global section reference count.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE
[SYS$LDR]SYS$VM.STB
[SYS$LDR]SYSTEM_PRIMITIVES.EXE
[SYS$LDR]SYSTEM_PRIMITIVES.STB
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_MIN.STB

VSI Case Identifiers

Jira BO-707
Netsuite NS2318

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

6.14. Performance Degradation After Installing SYS V3.0 ECO Kit

Problem Description

The SYS V3.0 ECO kit included several synchronization fixes which prevented potential system crashes. However, the behavior of how these changes interacted together had an inadvertent side effect.

An optimization for handling multiple page address ranges during the creation or mapping of global sections was effectively disabled. Applications with a significant reliance on this global section behavior could experience a noticeable performance degradation.

The affected behavior has been adjusted to have the required synchronization but also re-enable the performance optimization. With the change in this kit, the performance of the various $CRMPSC services has been restored to the same level as before installing the SYS V3.0 ECO kit.

Images and/or Files Affected

[SYS$LDR]SYS$VM.EXE

[SYS$LDR]SYS$VM.STB

VSI Case Identifiers

Jira BO-1649
Netsuite NS6705

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

6.15. ANALYZE/CRASH Cannot Analyze System Crash Dump File

Problem Description

Attempting to analyze a crash dump file with **`ANALYZE/CRASH`** may fail with an SDA-E-UNEXPECTED message similar to:

```
%SDA-E-UNEXPECTED, unexpected "Unknown" LMB in file #1 at VBN 01400051
```

This condition occurs if a memory section written to a dump file was incompletely saved, for example if the space reserved for saving the crash dump file was too small or if writing the dump file was interrupted.

Rather than exit from **`ANALYZE/CRASH`**, SDA will now issue the message to describe that some of the dump file cannot be interpreted, but then continue execution to allow analysis of already successfully read portions of the dump file.

Images and/or Files Affected

[SYSLIB]SDA$SHARE.EXE

VSI Case Identifiers

Jira KT-194
Netsuite NS5745

Release Version of VSI OpenVMS That Will Contain This Change

The next VSI OpenVMS IA-64 release after V8.4-2L3.

7. Images or Files Replaced

[SYS$LDR]EXCEPTION.EXE

Image Name:	"EXCEPTION"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"

Link Date/Time:	29-AUG-2026 03:32:54.21
Image Checksum (MD5):	6B25C8D383FC4485CE5EB510F0A76B2F

[SYS$LDR]EXCEPTION.STB

File Creation Date and Time:	29-AUG-2026 03:32:54.66
Checksum (MD5):	562A69E95448D1D7F909B715711F4387

[SYS$LDR]EXCEPTION_MON.EXE

Image Name:	"EXCEPTION_MON"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:32:54.82
Image Checksum (MD5):	957A86211F178D96F9E702AA08C28C62

[SYS$LDR]EXCEPTION_MON.STB

File Creation Date and Time:	29-AUG-2026 03:32:54.86
Checksum (MD5):	286D818DF3FB8BFED4D6AB3AE3D4DBEE

[SYS$LDR]PROCESS_MANAGEMENT.EXE

Image Name:	"PROCESS_MANAGEMENT"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:33:16.26
Image Checksum (MD5):	889A7B10A9C9677BB5CB947ABA102534

[SYS$LDR]PROCESS_MANAGEMENT.STB

File Creation Date and Time:	29-AUG-2026 03:33:18.64
Checksum (MD5):	A09302C5D4A64DA4EF2E8A72CFCDCC21

[SYS$LDR]PROCESS_MANAGEMENT_MON.EXE

Image Name:	"PROCESS_MANAGEMENT_MON"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"

Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:33:18.97
Image Checksum (MD5):	DDAC425F8B0D0B07F3241EA825A0BCD6

[SYS$LDR]PROCESS_MANAGEMENT_MON.STB

File Creation Date and Time:	29-AUG-2026 03:33:19.36
Checksum (MD5):	722998698A6ADADC524E5F2CD9A86201

[SYS$LDR]SECURITY.EXE

Image Name:	"SECURITY"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:32:56.43
Image Checksum (MD5):	522FC6254803A700FF4A085178CA19F4

[SYS$LDR]SECURITY.STB

File Creation Date and Time:	29-AUG-2026 03:32:56.87
Checksum (MD5):	5A7970C4E46F30BB5133C87CFB55CF23

[SYS$LDR]SECURITY_MON.EXE

Image Name:	"SECURITY_MON"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:32:57.20
Image Checksum (MD5):	9F922D1B1C6C0758D38AC1635E1D966E

[SYS$LDR]SECURITY_MON.STB

File Creation Date and Time:	29-AUG-2026 03:32:57.25
Checksum (MD5):	8D011871C696B61380E7099F45DEAD04

[SYS$LDR]SYS$VM.EXE

Image Name:	"SYS$VM"
Image File Identification:	"X-5"

Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:33:05.16
Image Checksum (MD5):	65BF7BE9F6E38BB5E36DBBF682B3C176

[SYS$LDR]SYS$VM.STB

File Creation Date and Time:	29-AUG-2026 03:33:09.08
Checksum (MD5):	413824F2D19E5A0A463D8B00B5F556C6

[SYS$LDR]SYSTEM_PRIMITIVES.EXE

Image Name:	"SYSTEM_PRIMITIVES"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:33:13.98
Image Checksum (MD5):	4D5DBC415BFFDB13F6853E3B43B03A9D

[SYS$LDR]SYSTEM_PRIMITIVES.STB

File Creation Date and Time:	29-AUG-2026 03:33:17.56
Checksum (MD5):	2040615970E5E9289276D56BC6646D8D

[SYS$LDR]SYSTEM_PRIMITIVES_MIN.EXE

Image Name:	"SYSTEM_PRIMITIVES_MIN"
Image File Identification:	"X-5"
Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:33:18.30
Image Checksum (MD5):	B507D1D1BB5053041CA26FC8CF28DB6E

[SYS$LDR]SYSTEM_PRIMITIVES_MIN.STB

File Creation Date and Time:	29-AUG-2026 03:33:18.79
Checksum (MD5):	60C9AE7C56ED3D6BD5EB42E2A9114061

[SYSLIB]SDA$SHARE.EXE

Image Name:	"SDA$SHARE"
Image File Identification:	"X-2"

Image Build Identification:	"XE4H-H4N-000211"
Link Identification:	"Linker I02-37"
Link Date/Time:	29-AUG-2026 03:53:26.84
Image Checksum (MD5):	297B93361A25B834DD86335B954A032E

Note

VMS Software, Inc. will only distribute kits in signed form. There is no need for most customers to compare file checksums for security or kit integrity reasons.

However, some sites may require such checking even when using signed kits. The image or file checksums (in MD5 format) are supplied to provide comparisons to the extracted final kit files. To find a file checksum, use:

```
$ CHECKSUM/ALGORITHM=MD5 filename
$ SHOW SYMBOL CHECKSUM$CHECKSUM
```

Note

Because a file or image may be replaced by multiple ECO kits over time, a PCSI generation number is used to ensure that the latest version of the file or image is preserved on your system during **PRODUCT INSTALL** of an ECO kit. Should a particular kit installation discover a newer version of a file or image in place on the system disk, the following message will be displayed:

```
%PCSI-I-RETAIN, file filename will not be replaced because file from kit
 has lower generation number
```

This is a normal occurrence depending on the order of kit installation. The correct version of the file or image will remain on the system after the current kit installation. The `%PCSI-I-RETAIN` message is informational only and does not indicate a problem.

8. Installation Instructions

8.1. Compressed File

This kit is provided for download within a ZIP archive container file.

The kit files may be extracted on any system with UNZIP and copied to your OpenVMS system, or extracted on your OpenVMS system directly.

Assuming you have created an UNZIP symbol to invoke the UNZIP image, you can invoke UNZIP to unpack the kit on OpenVMS using the command:

```
$ UNZIP VMS842L1I_SYS-V0500
```

This will extract the installable PCSI product kit file and its associated signed manifest (_VNC file), used for kit validation during **PRODUCT** commands.

VSI strongly recommends always using the manifest to validate the kit content during any **PRODUCT** commands. This will occur automatically if the files are both contained in the same directory.

UNZIP Tool Availability

Most customers likely have already installed a set of ZIP and UNZIP tools on their VSI OpenVMS systems. Should you need these tools, a set of the Info-ZIP freeware ZIP and UNZIP tools for VSI OpenVMS is available for download on the web at this address: https://vmssoftware.com/community/freeware/.

8.2. Installation Command

Install this kit with the POLYCENTER Software Installation Utility by logging into the SYSTEM account and typing the following command at the DCL prompt:

```
$ PRODUCT INSTALL VMS842L1I_SYS [/SOURCE=location_of_kit]
```

The kit location may be a tape drive, CD/DVD, or a disk directory that contains the kit. The **/SOURCE** qualifier is not needed if the **PRODUCT INSTALL** command is executed from the same directory as the kit location.

This kit requires the use of **/RECOVERY_MODE** and **/SAVE_RECOVERY_DATA** and will automatically set them; they do not need to be present on the command line.

The release notes for any kit may be extracted prior to kit installation using the **PRODUCT EXTRACT RELEASE_NOTES** command.

User-selectable options for installation behavior and scripting are available in this kit, refer to *Appendix A, "User-Selectable Control Options and Scripting Considerations"* for further details.

Additional help on installing PCSI kits can be found by typing **HELP PRODUCT INSTALL** at the system prompt.

9. Copyright

VMS SOFTWARE, INC. CONFIDENTIAL. This software is confidential proprietary software licensed by VMS Software, Inc., and is not authorized to be used, duplicated, or disclosed to anyone without the prior written permission of VMS Software, Inc.

Copyright 2026 VMS Software, Inc.

10. Disclaimer of Warranty and Limitation of Liability

This patch is provided as is, without warranty of any kind. All express or implied conditions, representations, and warranties, including any implied warranty of merchantability, fitness for particular purpose, or non-infringement, are hereby excluded to the extent permitted by applicable law. In no event will VMS Software, Inc. be liable for any lost revenue or profit, or for special, indirect, consequential, incidental or punitive damages, however caused and regardless of the theory of liability, with respect to any patch made available here or to the use of such patch.

11. Patch ID

I64VMS-VMS842L1I_SYS-V0500--4

Note

The terms "ECO kit" and "patch kit" may be used interchangeably in this document. This also applies for other VSI OpenVMS documentation when describing PCSI kits that provide remedial updates to a particular product.

A. User-Selectable Control Options and Scripting Considerations

A.1. Controlling Kit Behavior for Introductory Questions

This kit provides user-selectable control options for kit dialogue interaction and automated scripting capability as described here in this appendix.

The general form of a VSI OpenVMS ECO kit, when using **`PRODUCT INSTALL`**, consists of three initial questions regarding these topics:

1. System disk backup: A reminder that VSI recommends backing up the system disk before installing updates, followed by a `Do you want to continue?` YES/NO question, default is `YES`.
2. Reboot requirement: A summary of whether the kit being installed requires a system reboot, followed by a `Do you want to continue?` YES/NO question, default is `YES`.
3. Archival of updated files: A description of saving an "_OLD" copy of each image or file updated by the kit, followed by a `Do you want to save "_OLD" copies of replaced files?` YES/NO question, default is `NO`.

Other questions may be asked later, depending on the target disk or system environment or other kit-specific requirements.

Note

An initial `Do you want to continue?` question may be asked directly by the PCSI utility during any **`PRODUCT`** command—this has nothing to do with the kit being used. To avoid that question, you must supply sufficient detail to uniquely identify the product you wish to use and specify **`/OPTIONS=NOCONFIRM`** on the **`PRODUCT`** command.

Control options are available to customize the dialogue for the initial three kit questions. The controls are logical names, which may be defined in the process logical name table with a value of YES or NO.

To modify the behavior of a VSI OpenVMS ECO kit regarding the initial questions, define one or more of the following logical names before issuing the **`PRODUCT INSTALL`** command.

- To skip one or more of the questions, define the corresponding logical name shown here to YES:

SKIP$BACKUP	Skips system backup awareness question.
SKIP$REBOOT	Skips system reboot awareness question.
SKIP$ARCHIVE_OLD	Skips question about saving "_OLD" files. This will take the default, which is NO.

SKIP$INTRO	Skips all three of the above questions.

- To specifically override the default for saving "_OLD" files, define this logical name to YES or NO:

VSIKIT$ARCHIVE_OLD	Sets an answer for saving "_OLD" files behavior. This will skip the archive "_OLD" files question regardless of the above SKIP$* logical names.

- Two additional logical names may be defined as YES to modify the amount of explanatory text displayed for each question:

VSIKIT$VERBOSE	Shows all explanatory text for questions.
VSIKIT$BRIEF	Skips some general details in the explanations.

The default if neither name is defined is VERBOSE. If both names are defined to YES, VERBOSE overrides BRIEF. The BRIEF form is displayed for any questions that are skipped.

For example, to skip all three questions but save an archive "_OLD" copy of each replaced file:

```
$ DEFINE VSIKIT$ARCHIVE_OLD YES
$ DEFINE SKIP$INTRO YES
$ PRODUCT INSTALL kitname
```

A.2. Standard Behavior for YES/NO Questions Asked During Kit Installation

Any YES/NO questions asked during kit installation now follow these rules:

1. **Ctrl/Y** issued while a question is being asked will force the current **`PRODUCT`** operation to terminate. This is completely safe to do while the initial three questions are being asked during **`PRODUCT INSTALL`** as no changes have yet been made to the target disk.

2. Some questions may ignore **Ctrl/Y** and ask for a specific answer (for example, if aborting the current operation may have side effects for the system). Additionally, note the following:

   - PCSI may trap **Ctrl/Y** directly for some **`PRODUCT`** operations.

   - **Ctrl/Y** may be disabled during some sensitive kit processing.

3. The default YES/NO answer is automatically chosen if a kit is installed from a batch job, unless explicitly overridden by a logical name that provides the particular value, such as VSIKIT$ARCHIVE_OLD.

A.3. Installing a Kit From a Batch Job

To install a kit from a batch job, you will need to fully qualify the kit name so PCSI will have enough information to select the kit without asking for confirmation. For example, to install this kit:

```
$ PRODUCT INSTALL VMS842L1I_SYS/VERSION=V5.0/OPTIONS=NOCONFIRM
```

If the kit is located in a directory other than the current default directory, you will also need to add the qualifier:

`/SOURCE=location_of_the_kit`

For a batch job, any YES/NO question will automatically select the default answer. Use the control logical names explained above to modify the behavior if necessary. For the system disk backup and reboot questions, the batch behavior is identical to the default. For the save "_OLD" files question, define the VSIKIT$ARCHIVE_OLD logical name to YES if you want to save copies of the files, since the batch default is NO.

A.4. Deprecated Logical Names From HPE ECO Kits

The three names listed below were used by older VSI OpenVMS ECO kits for compatibility with HPE ECO kit behavior. These old names continue to function, but VSI encourages you to modify any scripts you may have to use the new names shown instead:

Old Name	**New Name**
NO_ASK$BACKUP	SKIP$BACKUP
NO_ASK$REBOOT	SKIP$REBOOT
ARCHIVE_OLD	VSIKIT$ARCHIVE_OLD