V9.2-2 Update V1.0 ECO Kit for VSI OpenVMS x86-64

Release Notes

Operating System: VSI OpenVMS x86-64 V9.2-2

Kit Name: VMS922X_UPDATE-V0100

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

Table of Contents

1. Kit Name

VMS922X_UPDATE-V0100

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 attached Disclaimer of Warranty and Limitation of Liability Statement.

2.2. Reboot Requirement

A reboot is mandated as part of installing this kit, performed automatically following the kit installation.

VSI OpenVMS for the x86 architecture uses a memory disk image, stored on the system disk, when booting the system. The content of the memory disk must remain consistent with the system disk content.

This kit updates the memory disk image and invokes a system reboot sequence (shutdown with reboot) directly as part of the kit installation. You must be prepared to allow the system reboot when installing the kit. After all other kit actions are complete, the system will automatically shutdown and reboot.

If you allow the reboot, you will have the choice of whether to invoke the site-specific shutdown procedure, SYS$MANAGER:SYSHUTDWN.COM during the shutdown portion of the reboot.

By default, after installation completes, the minutes until shutdown is zero. If you wish to leave additional time before the shutdown begins, define the system logical name SHUTDOWN $MINIMUM_MINUTES as the integer value of the wait time in minutes. For example:

```
DEFINE/SYSTEM SHUTDOWN$MINIMUM_MINUTES 10
```

No other options for the shutdown may be specified.

The shutdown will commence directly after the memory disk update as the final portion of the kit installation.

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

VSI OpenVMS x86-64 V9.2-2

2.4. Target Disk Requirements

This kit will create a new memory disk image file during **PRODUCT INSTALL** or **PRODUCT UNDO PATCH** operations. This file is required for x86 systems bootstrap. If this file cannot be correctly created, the target disk will not be bootable and you will need to restore from backup.

During **PRODUCT INSTALL**, the kit will check for sufficient space on the target disk. The minimum free space is 600,000 blocks to ensure that the memory disk image file and kit files can safely fit on the disk.

Additionally, a check is made to determine if the disk is too fragmented to correctly create the memory disk file. If either check fails, the installation will be aborted before making any changes. After you take corrective actions the **PRODUCT INSTALL** can be re-attempted.

These same checks are not automatically handled by the kit before a **PRODUCT UNDO PATCH** operation. This is not a frequent operation for customer systems. Should you need to remove this kit, you should ensure sufficient disk space before you start. The same 600,000 block minimum applies. To check fragmentation requirements, you can use the following commands:

```
$ ANALYZE/DISK/EXTENTS/REQUIRED=200000/NOOUTPUT disk
$ SHOW SYMBOL ANALYZE$REQUIRED_EXTENTS
```

If the symbol value is 25 or more, the disk is too fragmented and you should defragment it before using **PRODUCT UNDO PATCH**. You can accomplish this using a defragmentation tool or by restoring an image backup.

3. Kits Superseded by This Kit

VMS922X_HOTFIX-V0300	This HOTFIX kit contained the corrected images for the problems described in Sections 5.1 and 5.2 below. It was only distributed to the customer reporting the issues.

4. Kit Dependencies

VMS922X_PCSI-V0100

Note

The VMS922X_PCSI-V0100 kit must be installed prior to installing this kit, using a separate **PRODUCT INSTALL** command. The kits may not be jointly installed with a single PRODUCT INSTALL operation.

5. Problems Addressed in This Kit

5.1. System May Crash With DELCONPFN Bugcheck

Problem Description

When mapping large process-based PFN-mapped sections, the process working set may be left in an inconsistent state. When the process logs out, the system could then crash with a DELCONPFN bugcheck.

This issue can often occur when installing large images with the **/SHARE=ADDRESS/RESIDENT** qualifiers.

The problem is corrected with this patch kit.

Images and/or Files Affected

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

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

VSI Case Identifier

Jiras SPS-1273, SPS-1285, BO-1473

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.2. SYS$IO_PERFORM May Return Incorrect Data or Corrupt Files

Problem Description

Under some circumstances, the SYS$IO_PERFORM[W] system services may incorrectly compute the data buffer address for the I/O operation.

This can result in file corruption or application issues due to reading the wrong data.

The problem is corrected with this patch kit.

Images and/or Files Affected

[SYS$LDR]IO_ROUTINES.EXE
[SYS$LDR]IO_ROUTINES.STB
[SYS$LDR]IO_ROUTINES_MON.EXE
[SYS$LDR]IO_ROUTINES_MON.STB

VSI Case Identifier

Jiras SPS-1256, BO-1485

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.3. Disallow Shadow Sets as Dump Off System Disk (DOSD) Devices

Problem Description

The **SET DUMP/DEVICE** command allows the specification of shadow sets as Dump Off System Disk (DOSD) devices and **SHOW DUMP** will display a dump file found on the shadow set. However, the dump kernel cannot locate the dump file on a shadowed DOSD device.

This change causes SET DUMP/DEVICE to disallow shadow sets as DOSD devices.

Note

This restriction applies to DOSD devices only. The dump kernel can successfully write a crash dump to a shadowed system disk.

Images and/or Files Affected

[SYSEXE]SETSHOWDUMP.EXE

VSI Case Identifier

Jira KT-191

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

Workaround

Do not specify shadow sets to **`SET DUMP/DEVICE`**.

5.4. CLUE$STARTUP Exhausts System Disk Space if no DOSD Devices Are Mounted

Problem Description

An error in CLUE$STARTUP failed to end the search for a dump disk if none of the devices specified in the SET DUMP command were mounted at the time CLUE$STARTUP runs during boot.

This could result in an infinite loop within the procedure, and the resulting log file would eventually consume all the space on the system device.

This problem is corrected with this patch kit.

Images and/or Files Affected

[SYS$STARTUP]CLUE$STARTUP.COM

VSI Case Identifier

Jiras KT-188, KT-195

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

Workaround

Make sure any DOSD devices are mounted before CLUE$STARTUP runs. Mounting the devices in SYS $STARTUP:SYPAGSWPFILES.COM fulfills this condition.

5.5. Protected Shareable Image Issue Could Cause Process or System Crash

Problem Description

For a merge-activation of a protected shareable image, the transfer code to relocate the image was incorrectly allocated in P0 space. This can result in an access violation in an inner mode if the P0 address space is deleted or overwritten, causing a process or system crash.

With this patch kit, the transfer code is correctly allocated in a protected area in P1 space, eliminating the risk and potential crash.

Images and/or Files Affected

[SYS$LDR]SYSTEM_PRIMITIVES_0.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_0.STB
[SYS$LDR]SYSTEM_PRIMITIVES_0_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_0_MIN.STB
[SYS$LDR]SYSTEM_PRIMITIVES_2.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_2.STB
[SYS$LDR]SYSTEM_PRIMITIVES_2_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_2_MIN.STB
[SYS$LDR]SYSTEM_PRIMITIVES_3.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_3.STB
[SYS$LDR]SYSTEM_PRIMITIVES_3_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_3_MIN.STB
[SYS$LDR]SYSTEM_PRIMITIVES_4.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_4.STB
[SYS$LDR]SYSTEM_PRIMITIVES_4_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_4_MIN.STB
[SYS$LDR]SYSTEM_PRIMITIVES_6.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_6.STB
[SYS$LDR]SYSTEM_PRIMITIVES_6_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_6_MIN.STB
[SYS$LDR]SYSTEM_PRIMITIVES_7.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_7.STB
[SYS$LDR]SYSTEM_PRIMITIVES_7_MIN.EXE
[SYS$LDR]SYSTEM_PRIMITIVES_7_MIN.STB

VSI Case Identifier

Jira BO-1537

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

Workaround

Install the protected shareable image with shared address data.

5.6. RUNOFF Command May Display XPORT Errors or Corrupted Filenames

Problem Description

The RUNOFF code uses some special RMS internals to adjust RMS data structures for the files it is processing, for example the NAM fields using the FAB$V_NAM flag.

However, it incorrectly misses a few fields, especially the NAM$L_DEV field, which causes internal structure corruption when run on x86. This would result in XPORT errors and possibly display corrupted file names.

This problem is corrected with this patch kit.

Images and/or Files Affected

[SYSEXE]DSRINDEX.EXE
[SYSEXE]DSRTOC.EXE
[SYSEXE]RUNOFF.EXE

VSI Case Identifier

Jiras BO-1504, SPS-926, SPS-991

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.7. Various Corrections to OpenVMS Debugger Behavior

Problem Description

The following issues in the Debugger are corrected with this patch kit:

- Debugger is unable to examine Pascal strings.
- Debugger does not handle reading the DWARF namespace tag.
- Debugger cannot load a C++ program, failing with the following messages:

```
%DEBUG-E-ERROR, debugger error or session corruption
%DEBUG-I-INITIAL, Language: C++, Module: !AS
```

Images and/or Files Affected

[SYSLIB]DEBUG.EXE
[SYSLIB]DEBUGSHR.EXE

VSI Case Identifier

Jiras DEV-1683, DEV-1941, DEV-1967

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.8. ACMS Failures From Image Activation Issues

Problem Description

ACMS on x86 would encounter various failures, usually resulting in the server process crashing. Even the IVP would fail during product installation.

ACMS uses merge-activation of a protected shareable image. One portion of the structures involved in this mechanism was not correctly made permanent in P1 space. If the space was reused ACMS would encounter spurious failures.

The problem is corrected with this patch kit.

Images and/or Files Affected

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

VSI Case Identifier

Jira BO-1536

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.9. Misconfigured LAN Failover Devices or VLAN Devices for Cluster Boot

Problem Description

When a system configured as a cluster member boots, the cluster port driver, PEDRIVER, configures on all available LAN devices, including LAN Failover (LL) devices and VLAN (VL) devices.

For x86, a standalone version of LANACP.EXE is run to read the LAN device database and configure LAN Failover and VLAN devices.

An error in the code would only allow configuration of LLA and VLA devices, regardless of the actual device names. If the hardware included other devices, they would not be configured. If the hardware didn't include LLA or VLA, LANACP would attempt to configure the devices as LLA and VLA, resulting in an incorrect and unusable device configuration.

This problem is corrected with this patch kit.

Images and/or Files Affected

[SYSEXE]LANACP.EXE
[SYSEXE]LANACP_BFS.EXE

VSI Case Identifier

Jira SPS-1352

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

Workaround

This workaround requires redundant cluster communications paths so that cluster communications are not lost during these manual steps:

1. Remove the LAN Failover and VLAN definitions from the LAN permanent device database.
2. Reboot the system.
3. Do SCACP STOP LAN devname for the LAN Failover devices.
4. Do LANCP commands to set up the LAN Failover sets and VLAN devices.
5. Finally, do SCACP START LAN devname to start NISCA on the newly setup devices.

5.10. XA Gateway Failures in XaOpen() and IVP

Problem Description

XA Gateway XaOpen() function failed due to a latent coding error which did not affect other architectures for OpenVMS pre-x86.

Typical errors were like this one displayed by the IVP during kit installation:

```
IVP fail: failed to call gateway XaOpen function (-3)
```

This issue is corrected with this patch kit.

Images and/or Files Affected

[SYSEXE]DDTM$XG_SERVER.EXE

VSI Case Identifier

Jira SPS-1271

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.11. Add Support for Packed Decimal Routines Needed for BASIC

Problem Description

The following routines were added to the Object Time System (OTS) to enable packed decimal use required by the BASIC compiler.

- OTS$ADDP
- OTS$CMPP
- OTS$ASHP

These emulate the VAX instructions ADDPx, CMPPx and ASHP.

Images and/or Files Affected

[SYSLIB]LIBOTS2.EXE

VSI Case Identifier

Jiras RTLS-42, RTLS-424, DEV-2015

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.12. Availability Manager (DECamds) Could Induce INCONSTATE Bugcheck

Problem Description

The system will crash with an INCONSTATE bugcheck if the Availability Manager (DECamds) connects to an OpenVMS x86 system and displays single process information.

The routine used on x86 to collect process memory information for Availability Manager had an issue which could result in the INCONSTATE system crash.

This has been corrected with this patch kit.

Images and/or Files Affected

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

VSI Case Identifier

Jira AM-221

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.13. Access Violation in DTI$SHARE May Cause Process Deletion

Problem Description

When the DTI$SHARE image was installed with shared address data, a bug in the generated code caused an executive mode access violation. This would usually result in an unexpected process termination.

The issue is corrected with this patch kit.

Images and/or Files Affected

[SYSLIB]DTI$SHARE.EXE

VSI Case Identifier

Jiras SPS-1271, BO-1535

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

Workaround

Install the DTI$SHARE image without shared address data. On a running system which (by default) already has this image installed, use the command:

```
$ INSTALL REPLACE SYS$SHARE:DTI$SHARE - /OPEN/HEADER/SHARED=NOADDR/PROTECTED
```

5.14. Correct Spurious Exception Error Messages During Traceback Processing

Problem Description

Under some circumstances, traceback processing could encounter an exception, generating the message:

```
TRACEBACK - Exception occurred during traceback processing
```

An example of this occurred when the LSE editor encountered an error and attempted to display a traceback detailing the call stack.

This problem is corrected with this patch kit.

Images and/or Files Affected

[SYSLIB]TRACE.EXE

VSI Case Identifier

Jiras SPS-1339, DEV-2013

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.15. Add Support for x86 to BASIC Run-Time Library

Problem Description

Various modules in the BASIC RTL have been enhanced to enable x86-64 support. This includes support to:

- Extend some arguments from 32-bit integers to 64-bit integers
- Change conditional compilation symbols

- Update packed-decimal floating point passing mechanisms.

Images and/or Files Affected

[SYSLIB]DEC$BASRTL.EXE

VSI Case Identifier

Jira RTLS-423

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.16. DCL Procedure Error DOUBLEDEALO Bugcheck and Process Deletion

Problem Description

A DCL command procedure with certain syntax errors may result in a supervisor mode DOUBLEDEALO bugcheck, resulting in the deletion of the executing process.

This issue is corrected with this patch kit.

Images and/or Files Affected

[SYSEXE]DCL.EXE

VSI Case Identifier

Jiras SPS-1428, SPS-1429, BO-1541

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.17. Executive Mode Bugcheck May Cause KRNLSTAKNV System Crash

Problem Description

An executive mode bugcheck may be declared as fatal or non-fatal. A fatal executive mode bugcheck will crash the entire system. A non-fatal executive mode bugcheck will terminate the offending process.

If the BUGCHECKFATAL system parameter is set to 1, any executive mode bugcheck will cause a system crash (typically, this is used to get a crash dump snapshot of the underlying problem).

An error in the return mechanism from executive mode exception handling could result in a KRNLSTAKNV system crash, even if the BUGCHECKFATAL system parameter was set to 0 and the triggering executive mode bugcheck was declared as non-fatal.

This problem has been corrected with this patch 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 Identifier

Jiras SPS-1375, SPS-1343

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.18. Remove Spurious Linker Warnings

Problem Description

The linker sometimes reports conflicting attributes for .text sections.

Code sections in object modules should have the SHR attribute. The x86 compilers did not always set this attribute. With some .text sections being SHR and some NOSHR, the linker reports this with a warning. The warnings at link time are propagated into the image.

With this change, the linker silently changes all PSECTS with a name prefix of .text and the EXE attribute to SHR. This way no link time warnings are reported or propagated into the image file due to this condition.

Images and/or Files Affected

[SYSEXE]LINK.EXE

VSI Case Identifier

Jiras DEV-1944, DEV-2019

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.19. Improve SECURITY_SERVER Behavior for Systems Under Persistent Attack

Problem Description

Systems under persistent attack could fail to run the CIA database maintenance routine indefinitely. This could result in accumulating expired records over time, with the possibility of introducing transaction timeouts and denial of service. Should the database grow large enough, it could exhaust required resources and halt transaction processing entirely.

To alleviate this condition, a new server thread has been added which runs the CIA database periodic maintenance routine at a selectable interval regardless of attack volume. This guarantees that expired records are removed from the database and avoids the potential issues.

A new system logical name, SYS$CIA_MAINTENANCE_INTERVAL, now allows system managers to set the interval, in minutes, between maintenance runs. The interval can be set based on the attack volume for a system or cluster. The larger the attack volume, the smaller the interval should be to help keep the database and transaction times as small as possible. Intervals can be set from 1 to 60 minutes, with 10 minutes being the default.

For example, to set an interval of 5 minutes, use the command:

```
$ DEFINE/EXEC/SYSTEM SYS$CIA_MAINTENANCE_INTERVAL 5
```

Images and/or Files Affected

[SYSEXE]SECURITY_SERVER.EXE

VSI Case Identifier

Jira BO-1506

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.20. Various Fixes to SECURITY_SERVER Behavior

Problem Description

The following issues have been corrected in the security server with this patch kit:

- The server was adding errant equivalent entries on multi-node clusters where login attempts from the same remote source fail on multiple nodes.
- Under some circumstances, attempting to send a message to OPCOM could errantly return SS $_INVARG status. This prevented certain messages being sent to the console. It could also trigger an unnecessary restart of the server.
- Redefining a default proxy to a standard proxy could result in a request timeout.
- An ill-formed audit (INSFARG) is triggered when converting a default proxy to a standard proxy.

Images and/or Files Affected

[SYSEXE]SECURITY_SERVER.EXE

VSI Case Identifier

Jiras BO-1506, BO-1531

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

5.21. Some Exceptions May Mistakenly Crash the System

Problem Description

Some exceptions which should have only deleted the current process mistakenly crashed the system.

This problem is corrected with this patch 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 Identifier

Jiras BO-1540, SPS-1428

Release Version of VSI OpenVMS That Will Contain This Change

Next VSI OpenVMS x86-64 release after V9.2-2.

6. Problems Addressed from Previous Kits

None

7. Images or Files Replaced

[SYS$LDR]EXCEPTION.EXE

Image name	"EXCEPTION"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:51.58
Image Checksum (MD5)	E8C8E72C54BFCFF5326A545C1CD52C3B

[SYS$LDR]EXCEPTION.STB

File creation date and time	12-FEB-2024 07:38:54.47
Checksum (MD5)	4D6EF337E286054F9B3ED19DD41D970A

[SYS$LDR]EXCEPTION_MON.EXE

Image name	"EXCEPTION_MON"

Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:58.05
Image Checksum (MD5)	8AB46C081BC860A747BAAA36B595CB69

[SYS$LDR]EXCEPTION_MON.STB

File creation date and time	12-FEB-2024 07:38:58.40
Checksum (MD5)	E541E8431C1DA42C79EA5D286F1E2218

[SYS$LDR]IMAGE_MANAGEMENT.EXE

Image name	"IMAGE_MANAGEMENT"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:03.95
Image Checksum (MD5)	274E1246AA50B383B2478F450B200C8E

[SYS$LDR]IMAGE_MANAGEMENT.STB

File creation date and time	12-FEB-2024 07:39:04.14
Checksum (MD5)	4A6CFB743AD2752B5439B9B0EE270E0E

[SYS$LDR]IO_ROUTINES.EXE

Image name	"IO_ROUTINES"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:05.40
Image Checksum (MD5)	1D678962EA2FDB4F0C8417979A20D8EF

[SYS$LDR]IO_ROUTINES.STB

File creation date and time	12-FEB-2024 07:39:05.76
Checksum (MD5)	A4CE7983D53978E4297722B558388D9D

[SYS$LDR]IO_ROUTINES_MON.EXE

Image name	"IO_ROUTINES_MON"

Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:06.78
Image Checksum (MD5)	96A09E868229FD263B895E19660C32BB

[SYS$LDR]IO_ROUTINES_MON.STB

File creation date and time	12-FEB-2024 07:39:07.35
Checksum (MD5)	F508A1976E5AC714D4315FDADCE47F77

[SYS$LDR]SYS$VM.EXE

Image name	"SYS$VM"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:04.46
Image Checksum (MD5)	9BA4D6FFA955D8E37E8D84AB6716A2E2

[SYS$LDR]SYS$VM.STB

File creation date and time	12-FEB-2024 07:39:04.81
Checksum (MD5)	7B55736E311AB3AF494DA735C398F46E

[SYS$LDR]SYS$VM_MON.EXE

Image name	"SYS$VM_MON"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:05.80
Image Checksum (MD5)	77B3BFD4D1E65150CDB516C6CCFA093E

[SYS$LDR]SYS$VM_MON.STB

File creation date and time	12-FEB-2024 07:39:06.25
Checksum (MD5)	010719D28488D1908E727110C9E4E8C9

[SYS$LDR]SYSTEM_PRIMITIVES_0.EXE

Image name	"SYSTEM_PRIMITIVES_0"

Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:10.13
Image Checksum (MD5)	8932BEF9D6EBD137AF0359A8299901E5

[SYS$LDR]SYSTEM_PRIMITIVES_0.STB

File creation date and time	12-FEB-2024 07:39:10.46
Checksum (MD5)	F72BF4D5AF126690C8D058A2E4572006

[SYS$LDR]SYSTEM_PRIMITIVES_0_MIN.EXE

Image name	"SYSTEM_PRIMITIVES_0_MIN"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:10.90
Image Checksum (MD5)	33845D6B4A5DB97B122C515D563ABF74

[SYS$LDR]SYSTEM_PRIMITIVES_0_MIN.STB

File creation date and time	12-FEB-2024 07:39:11.17
Checksum (MD5)	79673062DA2BE7A31698037DBF7CB59D

[SYS$LDR]SYSTEM_PRIMITIVES_2.EXE

Image name	"SYSTEM_PRIMITIVES_2"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:11.65
Image Checksum (MD5)	7A1BC23BB57552F3FAC71520FEEECBE1

[SYS$LDR]SYSTEM_PRIMITIVES_2.STB

File creation date and time	12-FEB-2024 07:39:11.84
Checksum (MD5)	2E3F58A0AA785FA448DE1CE75E56E028

[SYS$LDR]SYSTEM_PRIMITIVES_2_MIN.EXE

Image name	"SYSTEM_PRIMITIVES_2_MIN"

Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:12.16
Image Checksum (MD5)	CBAA6B683A913722B7B67FE347087477

[SYS$LDR]SYSTEM_PRIMITIVES_2_MIN.STB

File creation date and time	12-FEB-2024 07:39:12.40
Checksum (MD5)	7B9E808AB66564D5E32C22D9E056CAD6

[SYS$LDR]SYSTEM_PRIMITIVES_3.EXE

Image name	"SYSTEM_PRIMITIVES_3"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:12.74
Image Checksum (MD5)	23C9D61D414D2491FC7547C4F051ED50

[SYS$LDR]SYSTEM_PRIMITIVES_3.STB

File creation date and time	12-FEB-2024 07:39:12.93
Checksum (MD5)	147EFE3F6212D383A476382E599FBABB

[SYS$LDR]SYSTEM_PRIMITIVES_3_MIN.EXE

Image name	"SYSTEM_PRIMITIVES_3_MIN"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:13.21
Image Checksum (MD5)	A518C535A4A1C8E8CC24F7A9674A94EC

[SYS$LDR]SYSTEM_PRIMITIVES_3_MIN.STB

File creation date and time	12-FEB-2024 07:39:13.40
Checksum (MD5)	53AE2BBF80873494404D452F96881B96

[SYS$LDR]SYSTEM_PRIMITIVES_4.EXE

Image name	"SYSTEM_PRIMITIVES_4"

Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:13.75
Image Checksum (MD5)	0FA8F8A793479317D5BF21565E0C0F90

[SYS$LDR]SYSTEM_PRIMITIVES_4.STB

File creation date and time	12-FEB-2024 07:39:14.04
Checksum (MD5)	94ABA1AAAFD8C3CC5D5C31D6A13DE5A3

[SYS$LDR]SYSTEM_PRIMITIVES_4_MIN.EXE

Image name	"SYSTEM_PRIMITIVES_4_MIN"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:14.55
Image Checksum (MD5)	736A571D8B46EFD54C758BCDFBF7CC50

[SYS$LDR]SYSTEM_PRIMITIVES_4_MIN.STB

File creation date and time	12-FEB-2024 07:39:14.73
Checksum (MD5)	91140EDB9402BA342E846362624B4B01

[SYS$LDR]SYSTEM_PRIMITIVES_6.EXE

Image name	"SYSTEM_PRIMITIVES_6"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:15.37
Image Checksum (MD5)	88733ED66B8B94BA9079EFA7EB2598FC

[SYS$LDR]SYSTEM_PRIMITIVES_6.STB

File creation date and time	12-FEB-2024 07:39:15.57
Checksum (MD5)	30E759D78ACA88FEF039FBD7100D75F0

[SYS$LDR]SYSTEM_PRIMITIVES_6_MIN.EXE

Image name	"SYSTEM_PRIMITIVES_6_MIN"

Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:15.89
Image Checksum (MD5)	EAB7EC21FED1AC1DDEFB8C257AF67A6C

[SYS$LDR]SYSTEM_PRIMITIVES_6_MIN.STB

File creation date and time	12-FEB-2024 07:39:16.08
Checksum (MD5)	A0700A2A069A8A8318FA484785A72557

[SYS$LDR]SYSTEM_PRIMITIVES_7.EXE

Image name	"SYSTEM_PRIMITIVES_7"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:16.39
Image Checksum (MD5)	23A66AAEA3EE9C06274783116AC56BC3

[SYS$LDR]SYSTEM_PRIMITIVES_7.STB

File creation date and time	12-FEB-2024 07:39:16.58
Checksum (MD5)	A751698969AE59A652087C7A8183DD2A

[SYS$LDR]SYSTEM_PRIMITIVES_7_MIN.EXE

Image name	"SYSTEM_PRIMITIVES_7_MIN"
Image file identification	"X-16"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:17.45
Image Checksum (MD5)	D52D32208069C5C9A2D15E32971B13D6

[SYS$LDR]SYSTEM_PRIMITIVES_7_MIN.STB

File creation date and time	12-FEB-2024 07:39:17.67
Checksum (MD5)	489EA64B5F9053EC2CDF5D3E3DFC15BA

[SYS$STARTUP]CLUE$STARTUP.COM

File creation date and time	16-NOV-2023 09:05:24.75
Checksum (MD5)	8D7AB0E9329BB815B4642195CFC7E908

[SYSEXE]DCL.EXE

Image name	"DCL"
Image file identification	"X-61"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:54.33
Image Checksum (MD5)	21E61DE67D60F088887ADA0D1116ED33

[SYSEXE]DDTM$XG_SERVER.EXE

Image name	"DDTM$XG_SERVER"
Image file identification	"V2.1-X035-XGLJ"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:03.32
Image Checksum (MD5)	3D5FE141616718CDCDD9825B078D775A

[SYSEXE]LANACP.EXE

Image name	"LANACP"
Image file identification	"X-37"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:09.40
Image Checksum (MD5)	97EC503434C7F8FA094F60ED4F1D24BF

[SYSEXE]LANACP_BFS.EXE

Image name	"LANACP_BFS"
Image file identification	"X-37"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:10.37
Image Checksum (MD5)	7FCA27DC7C3EAA3E5183E7C7C45EFCAF

[SYSEXE]LINK.EXE

Image name	"LINK"
Image file identification	"I02-97"
Image build identification	"XGLJ-P4A-000011"

Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:51.54
Image Checksum (MD5)	DBAEB8A701631E3E8D77AA507A5B2353

[SYSEXE]SECURITY_SERVER.EXE

Image name	"SECURITY_SERVER"
Image file identification	"X-4"
Image build identification	""
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:53.77
Image Checksum (MD5)	056DACDC0E67DDB035DF46619ECD3C8B

[SYSEXE]SETSHOWDUMP.EXE

Image name	"SETSHOWDUMP"
Image file identification	"X-14"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:54.77
Image Checksum (MD5)	3EBEEA3446887816F2E46A882EC45F16

[SYSEXE]DSRINDEX.EXE

Image name	"DSRINDEX"
Image file identification	"V3.2-01"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:00.36
Image Checksum (MD5)	D260B77E7517EFEBA50B9A9148F4B377

[SYSEXE]DSRTOC.EXE

Image name	"DSRTOC"
Image file identification	"V3.2-01"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:59.43
Image Checksum (MD5)	A14AE6934032C0C233F0531A834388F4

[SYSEXE]RUNOFF.EXE

Image name	"RUNOFF"
Image file identification	"V3.2-01"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:55.46
Image Checksum (MD5)	DA0B732E15C0C041135857C641C41757

[SYSLIB]DEBUG.EXE

Image name	"DEBUG"
Image file identification	"V9.2-009"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:39:01.81
Image Checksum (MD5)	3B29B822F46C2F660F95C30DA9A6E71B

[SYSLIB]DEBUGSHR.EXE

Image name	"DEBUGSHR"
Image file identification	"V9.2-009"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:53.29
Image Checksum (MD5)	D3B8869AD350915BFE32CA6937A5744B

[SYSLIB]DEC$BASRTL.EXE

Image name	"DEC$BASRTL"
Image file identification	"X01-039"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:24.03
Image Checksum (MD5)	39400F12B6D4EBD6CB3C87F2417F2A8F

[SYSLIB]DTI$SHARE.EXE

Image name	"DTI$SHARE"
Image file identification	"V2.1-X035-XGLJ"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"

Link Date/Time	12-FEB-2024 07:37:30.86
Image Checksum (MD5)	5869EFF245C8A59571434E1E69E5BD0D

[SYSLIB]LIBOTS2.EXE

Image name	"LIBOTS2"
Image file identification	"V2.8-77"
Image build identification	"XGLJ-P4A-000011"
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:08.07
Image Checksum (MD5)	548A680B5362CA65A8FDC126CA2C54ED

[SYSLIB]TRACE.EXE

Image name	"TRACE"
Image file identification	"V9.2-001"
Image build identification	""
Link identification	"Linker I02-94"
Link Date/Time	12-FEB-2024 07:38:13.04
Image Checksum (MD5)	76785B9DF3922D7E6A1EAA7F5741AFDB

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 are supplied (in MD5 format) 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 patch 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 a patch 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.

Info-ZIP's freeware ZIP and UNZIP tools are provided for use on this VSI OpenVMS version. Your site may have already set up symbols for these tools or other equivalent ZIP tools. If not, use the following command to define a symbol to run the UNZIP image:

```
$ UNZIP == "$SYS$SYSDEVICE:[VMS$COMMON.SYSHLP.UNSUPPORTED.UNZIP]UNZIP"
```

Then invoke UNZIP to unpack the kit using the command:

```
$ UNZIP VMS922X_UPDATE-V0200
```

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.

8.2. Installation Command

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

```
$ PRODUCT INSTALL VMS922X_UPDATE [/SOURCE=location_of_kit]
```

The kit location may be a 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.

8.3. Special Installation Instructions

Should you need to remove this kit via **PRODUCT UNDO PATCH**, the same mandated reboot requirement is in effect as the memory disk image is changed back to the prior system content.

The kit will update the memory disk image automatically as the final part of installation.
There is currently no mechanism within the PCSI utility to cleanly invoke a system reboot for **PRODUCT UNDO PATCH**.

You will be instructed as the kit exits that you must perform this function manually in this case.

Note

When the SYS$MD.COM procedure is executing to update the memory disk image, some errors may be reported similar to these:

```
%INSTALL-I-NONRESSHRADR, image installed ignoring '/RESIDENT' image_name
                -INSTALL-E-NOGHREG, insufficient memory in the code or data granularity
                 hint region
```

or

```
%INIT-F-GHREGIONFULL, An allocation was attempted from GH region GH_RES_CODE_S2 but
                there is not enough space in the region for the allocation.
```

These are due to having both old and new copies of some images which are still being used until the system is rebooted. Typically they may be ignored as they will clear up during the reboot. Should there still be similar messages during system startup after reboot, you may need to use AUTOGEN to adjust the related system parameters.

Note

During **PRODUCT INSTALL** or **PRODUCT UNDO PATCH**, the PCSI utility may issue the following message:

```
There is not enough space on the memory disk.
                You must take these steps now to complete this installation:

                -  Run @SYS$UPDATE:SYS$MD
                -  Reboot the system
```

In all cases the kit procedure will run SYS$MD, there is no need for you to do this.

For **PRODUCT INSTALL**, the reboot is also automatically handled by the kit procedure and you need not do a reboot yourself.

For **PRODUCT UNDO PATCH**, you must manually reboot the system after the operation completes. The kit dialogue will remind you of this requirement at the end of the operation. There is currently no mechanism in PCSI to automatically invoke the system shutdown and reboot for **UNDO PATCH** operations.

9. Copyright

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

X86VMS-VMS922X_UPDATE-V0200--4

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 ECO patch 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 patches, 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 `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 the VSI ECO patch 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, if aborting the current operation may have side effects. 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 which 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 VMS922X_UPDATE/VERSION=V2.0/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 Patch Kits

The three names listed below were used by older VSI OpenVMS patch kits for compatibility with HPE patch 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