

VSI Git V2.44-1A for OpenVMS

Release Notes

Publication Date: October 2024

Software Versions: VSI OpenVMS IA-64 Version 8.4-2L1 or higher
VSI OpenVMS x86-64 Version 9.2-1 or higher

Kit Names: VSI-I64VMS-GIT-V0244-1A01-1.PCSI
VSI-X86VMS-GIT-V0244-1A01-1.PCSI

Table of Contents

1. Introduction	3
2. Requirements	3
2.1. Software Requirements	3
2.2. Before You Begin	3
3. Related Documentation	4
4. Installing the Kit	4
5. Post-Installation Steps	5
6. Logical Names	5
7. Supported Commands	5
8. Known Issues and Limitations	6

1. Introduction

Git (<https://git-scm.com/>) is a popular open-source distributed version control system.

VSI Git for OpenVMS is a port of Git for the VSI OpenVMS operating environment. This port enables OpenVMS users to utilize Git source code management capabilities, which should facilitate distributed development, code versioning, and collaboration across teams.

VSI Git for OpenVMS is released under the GNU General Public License version 2 (GPLv2).

This version of VSI Git for OpenVMS is based on the Portable Git distribution available at <https://github.com/git/git/releases/tag/v2.44.0>.

2. Requirements

2.1. Software Requirements

VSI Git V2.44-1A for OpenVMS has been compiled and built using the operating system and compiler versions listed below. VSI does not recommend installing and using this kit on older versions.

- VSI OpenVMS IA-64 V8.4-2L1 or higher; note the following additional requirements:
 - On VSI OpenVMS IA-64 V8.4-2L1, the ECO VMS842L1I_RTL-V0600 kit or later is required.
 - On VSI OpenVMS IA-64 V8.4-2L3, the ECO VMS842L3I_RTL-V0600 kit or later is required.

Note

The RTL ECO kits mentioned above require their respective DPML V0200 ECO to be installed first.

- VSI OpenVMS x86-64 V9.2-1 or higher
- ODS-5 file system
- VSI SSL3 V3.0-14 or later
- VSI CURL V8.0-1A (this component is required only if you plan to use VSI Git for OpenVMS with the HTTP/HTTPS protocol; see Section 8, “Known Issues and Limitations”)
- VSI Vim for OpenVMS (only on IA-64)

Important

You do not need to install VSI Vim if you are using GNV. VSI Vim is a part of GNV and is located in the GNV/BIN directory.

The VIM.EXE file must be assigned the Owner:RWED privilege.

2.2. Before You Begin

Before you start using VSI Git for OpenVMS, consider the following:

- If more than one version of the same file is present in the `git` directory, the user must purge the directory before using the `git checkout`, `git merge`, `git reset`, or `git stash` commands.
- To ensure that the output of the `git branch` command is displayed correctly, enter the following command before using Git:

```
$ SET TERMINAL/INQUIRE
```

- To accept the extended format names, enter the following command:

```
$ SET PROCESS/PARSE_STYLE=EXTENDED
```

3. Related Documentation

Before installing and using VSI Git for OpenVMS, VSI recommends that you read the official documentation for open-source Git in order to better understand the software.

4. Installing the Kit

VSI Git V2.44-1A for OpenVMS is provided as an OpenVMS PCSI kit that a suitably privileged user can install by running the following command:

```
$ PRODUCT INSTALL GIT
```

Warning

Do not use the `/DESTINATION` qualifier with the `PRODUCT INSTALL` command when installing VSI Git for OpenVMS x86-64 to specify an alternative (non-default) installation location. Git components are bundled with the VSI OpenVMS for x86-64 operating system, which imposes specific requirements in terms of location of these components and all associated configuration files.

Your installation output will look similar to the following:

```
The following product has been selected:
  VSI X86VMS GIT V2.44-1A                Layered Product [Installed]

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.

Configuring VSI X86VMS GIT V2.44-1A: VSI GIT

  © Copyright 2024 VMS Software, Inc.

  VSI GIT is released under a BSD license, or a license more free than that.

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
  VSI X86VMS GIT V2.44-1A                DISK$XGLO:[VMS$COMMON.]
```

Portion done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

The following product has been installed:

VSI X86VMS GIT V2.44-1A Layered Product

VSI X86VMS GIT V2.44-1A: VSI GIT

Post installation tasks required to run VSI GIT.

To start VSI GIT at system boot time, add the following line to SYS\$MANAGER:SYSTARTUP_VMS.COM:

```
$ GITSTART := SYS$STARTUP:GIT$STARTUP.COM
$ IF f$search("'GITSTART'") .NES. "" THEN @'GITSTART'
```

To define the necessary VSI GIT symbols and set up the environment for using git commands, execute the following script:

```
$ @GIT$ROOT:[GIT_CORE]GIT$DEFINE_COMMANDS.COM
```

For more information, refer to this associated document:
- <https://vsi.atlassian.net/wiki/spaces/GIT/overview>

\$

5. Post-Installation Steps

After installing VSI Git for OpenVMS, you must define and assign several logical names and symbols that required to use the product. To do that automatically, enter the following commands in this exact order:

```
@ SYS$STARTUP:GIT$STARTUP
@git$root:[git_core]git$define_commands.com
```

6. Logical Names

VSI Git for OpenVMS uses the following logical names:

Logical Name	Description
GIT\$DISABLE_CASE_SENSITIVE_MODE	Disables the case-sensitive mode.
VIM\$ROOT	Vim installation directory.
GIT\$ROOT	Git installation directory.

7. Supported Commands

VSI Git V2.44-1A for OpenVMS supports the following Git commands:

- `git add`
- `git bisect`
- `git branch`
- `git checkout`

- `git clone`
- `git commit`
- `git fetch`
- `git init`
- `git log`
- `git merge`
- `git pull`
- `git push`
- `git reset`
- `git stash`
- `git status`

VSI OpenVMS supports the **HELP GIT** command, which brings up a list of supported Git commands and a list of issues and limitations applicable to the current release of VSI Git for OpenVMS.

8. Known Issues and Limitations

- An RPC error may occur when executing Git commands, typically due to network issues, working with large files, or limitations with Git configurations. The error message may look similar to the following:

```
error: RPC failed; curl 18 HTTP/2 stream 1 was not closed cleanly before end
  of
the underlying connection
error: 386 bytes of body are still expected
fatal: early EOFpected disconnect while reading sideband packet
```

To resolve this issue, you can utilize one of the following workarounds:

- Switch to SSH authentication (for example, `git@`).
- If the issue occurred when cloning a git repository, perform a shallow clone by using the `--depth` option with the `git clone` command.
- Using VSI Git for OpenVMS with the HTTP/HTTPS protocol requires the **Share** user privilege and VSI CURL V8.0-1A or higher.