

Lua 5.3.5 for OpenVMS I64 and Alpha

August 2020

1. Introduction

Thank you for your interest in this port of Lua to OpenVMS. The current release of Lua for OpenVMS is based on the Lua 5.3.5 Open Source distribution.

Lua (<http://www.lua.org>) is a powerful, fast, lightweight, embeddable scripting language that combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting byte-codes using a register-based virtual machine, and has automatic memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping tasks.

This OpenVMS port of Lua includes the core binary Lua distribution and several extensions, including a lightweight pure Lua JSON library and a set of OpenVMS-specific extensions to facilitate calling of various OpenVMS system service and RTL routines. It is anticipated that additional extensions will be included in future releases.

2. Requirements

The kit you are receiving has been compiled and built using the operating system and compiler versions listed below. While it is highly likely that you will have no problems installing and using the kit on systems running higher versions of the products listed, we cannot say for sure that you will be so lucky if your system is running older versions.

- OpenVMS 8.4-1H1 I64 or higher
- The software must be installed on an ODS-5 enabled disk (the installation will fail if this requirement is not met)
- A TCP/IP stack (if you wish to perform network operations)
- C compiler - HP C V7.1-015 (optional; required only if you intend to develop your own extensions)

In addition to the above requirements, it is assumed that the reader has a good knowledge of OpenVMS and of software development in the OpenVMS environment.

3. Recommended reading

It is recommended that developers read some of the excellent tutorials and other documentation available via the Lua web site (<http://www.lua.org/docs.html>). The web site provides numerous links to books, technical guides, whitepapers, and numerous other documents that provide useful information on developing applications using Lua.

4. Installing the kit

The kit is provided as an OpenVMS PCSI kit (VSI-I64VMS-LUA-V0503-5A-1.PCSI or VSI-AXPVMS-LUA-V0503-5A-1.PCSI, depending on platform) that can be installed by a suitably privileged user using the following command:

```
$ PRODUCT INSTALL LUA
```

The installation will then proceed as follows (output may differ slightly from that shown, depending on platform and other factors):

```
Performing product kit validation of signed kits ...
```

```
The following product has been selected:
```

```
VSI I64VMS LUA V5.3-5A          Layered Product
```

```
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 I64VMS LUA V5.3-5A: Lua for OpenVMS is based on Lua Version 5.3.5
```

```
© Copyright 2020 VMS Software Inc.
```

```
VSI Software Inc.
```

```
* This product does not have any configuration options.
```

```
Execution phase starting ...
```

```
The following product will be installed to destination:
```

```
VSI I64VMS LUA V5.3-5A          DISK$I64SYS:[VMS$COMMON.]
```

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

```
The following product has been installed:
```

```
VSI I64VMS LUA V5.3-5A          Layered Product
```

```
VSI I64VMS LUA V5.3-5A: Lua for OpenVMS is based on Lua Version 5.3.5
```

```
Post-installation tasks are required.
```

```
To start the Lua runtime at system boot time, add the following lines to SYS$MANAGER:SYSTARTUP_VMS.COM:
```

```
$ file := SYS$STARTUP:LUA$STARTUP.COM
$ if f$search("''file'") .nes. "" then @'file'
```

```
To stop Lua at system shutdown, add the following lines to SYS$MANAGER:SYSHUTDOWN.COM:
```

```
$ file := SYS$STARTUP:LUA$SHUTDOWN.COM
$ if f$search("''file'") .nes. "" then @'file'
```

4.1. *Post-installation steps*

After the installation has successfully completed, include the commands displayed at the end of the installation procedure into `SYSTARTUP_VMS.COM` to ensure that the logical names required in order for users to use the software are defined system-wide at start-up.

Users will then be able to use the Lua interpreter and the Lua bytecode compiler by defining foreign commands as follows:

```
$ LUA ::= $LUA$ROOT:[BIN]LUA.EXE
$ LUAC ::= $LUA$ROOT:[BIN]LUAC.EXE
```

Developers can implement their own dynamic modules by including in their projects the Lua header files found in `LUA$ROOT:[INCLUDE]` and by linking with either the shareable image `LUA$ROOT:[LIB]LUA$SHR.EXE` or the object library `LUA$ROOT:[LIB]LIBLUA.OLB`. Note that the logical name `LUA$SHR` is defined by the Lua start-up procedure and may be used to link with the shareable image. Symbols in the shareable image and object library are mixed-case, and developers should use the C compiler option `/NAMES=(AS_IS,SHORTENED)` or include in their code appropriate `#pragma` directives to ensure that symbols are correctly resolved when linking.

Generally speaking there are no special quota or privilege requirements required for users wishing to develop applications using Lua, although it should be noted that some extensions may have special requirements (for example, networking extensions may require a high `BYTLM` quota).

4.2. *Installing in an alternative location*

By default the software will be installed in `SYS$SYSDEVICE:[VMS$COMMON]`. If you wish to install the software in an alternative location this can be achieved using the `/DESTINATION` qualifier with the `PRODUCT INSTALL` command to specify the desired location; however it is important to note that an additional manual step will then be required to complete the installation. Specifically, when an alternative destination is specified, the start-up and shutdown procedures (`LUA$STARTUP.COM` and `LUA$SHUTDOWN.COM`) will be placed into a subdirectory `[.SYS$STARTUP]` residing under the specified destination directory. If you wish to run these files from your standard `SYS$STARTUP` directory they will need to be copied from the destination subdirectory into your systems `SYS$STARTUP` directory.

5. **Included modules**

As noted previously, this release of Lua for OpenVMS includes the core binary distribution and several extensions. The included additional modules are listed below.

- LuaFileSystem (<http://keplerproject.github.com/luafilename/>)
- JSON (<https://github.com/rxi/json.lua>)
- Lua-zip (<https://github.com/brimworks/lua-zip>)
- fifo (<https://github.com/daurnimator/fifo.lua>)
- binaryheap (<https://github.com/Tieske/binaryheap.lua>)
- basex (<https://github.com/aiq/basexx/>)
- LuaSocket (<https://github.com/diegonehab/luasocket>)

In addition to these modules, the kit also includes a set of extensions to facilitate calling of various OpenVMS system service and RTL routines. Refer to the various files provided in `LUA$ROOT:[EXAMPLES]` for examples of how to use some of these extensions. It is hoped that these extensions will be fully documented in future releases of the software.