



VMS Software

WebUI Version 4.2-0 for VSI OpenVMS IA-64 and x86-64

Release Notes

Publication Date: June 2025

Operating Systems: VSI OpenVMS IA-64 Version 8.4-2L1 or higher
VSI OpenVMS x86-64 Version 9.2-3 or higher

Software Version: WebUI Version 4.2-0 for VSI OpenVMS IA-64 and x86-64

Kit Names: VSI-I64VMS-WEBUI-V0402-0-1.PCSI
VSI-X86VMS-WEBUI-V0402-0-1.PCSI

Table of Contents

1. Introduction	3
2. What's New in This Release	3
3. Requirements	4
4. Before Installing	4
5. Installing the Kit	4
6. Post-Installation Steps	5
7. Privileges and Quotas	8
8. Installing in an Alternative Location	8
9. Python Language Bindings	9
10. Known Problems and Limitations	9

Note

If you are upgrading from a previous version of WebUI, be sure to upgrade both CivetWeb and Lua to the minimum required versions specified in these release notes before upgrading WebUI.

1. Introduction

Thank you for your interest in the WebUI for VSI OpenVMS IA-64 and x86-64. The current release of WebUI for VSI OpenVMS is 4.2-0.

The WebUI for VSI OpenVMS provides a RESTful interface for the management of VSI OpenVMS systems. In addition to the core REST API, the software includes a modern and intuitive web browser-based graphical user interface and Python language bindings that can be used to implement management tools and scripts that can be run from any platform to perform automated remote management tasks.

Core management functions supported by this release of the WebUI include:

- Management of users and rights identifiers
- Process management
- Management of batch and printer queues
- Viewing file system information, including various XFC data
- Monitoring of CPU and memory usage
- License management
- Management of system parameters
- Installed products (PCSI only)
- User-defined reports
- TCP/IP configuration management (VSI and HPE TCP/IP Services only)
- Management of logical names
- A WebSockets-based terminal interface (browser only)
- Examining error log messages

In addition, WebUI provides basic monitoring facilities to report events such as device errors and intrusions, and OpenVMS performance monitoring is provided via an optional plugin component for VSI PERFDAT.

2. What's New in This Release

This release of WebUI is provided as a licensed OpenVMS product and therefore requires a license to use. As with usual licensed products, register the product with the **LICENSE REGISTER** command.

Additionally, the WebSockets terminal interface port is now configurable (see [here](#) for details).

3. Requirements

The kit you are receiving has been built and tested using the operating system 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 operating system or products listed, we cannot guarantee functionality if your system is running older versions.

- VSI OpenVMS 8.4-2L1 or higher (IA-64) or VSI OpenVMS Version V9.2-3 or higher (x86-64)
- VSI TCP/IP or HPE TCP/IP Services for OpenVMS
- CivetWeb 1.17-0 for VSI OpenVMS or higher
- Lua 5.3-5H for VSI OpenVMS or higher (required by CivetWeb)

In addition to the above software product requirements, WebUI must be installed onto an ODS-5 enabled file system. It should also be noted that while WebUI can be used in conjunction with the MultiNet TCP/IP stack, the WebUI TCP/IP module will not function correctly with MultiNet.

Important

The SYSUAF logical name must be defined with a path to the SYSS\$UAF.DAT authorization file. Define the logical name with the following command:

```
$ DEFINE/SYSTEM SYSUAF SYS$SYSTEM:SYS$UAF.DAT
```

4. Before Installing

Before installing the WebUI as described in the following section, it is important to ensure that the CivetWeb web server is installed and is running, or at a minimum that the system logical name CIVETWEB\$ROOT is defined and points to a valid CivetWeb installation. The WebUI files are installed under the directory pointed to by this logical name. If this location is not defined or does not exist, then the WebUI installation will fail.

5. Installing the Kit

The kit is provided as an OpenVMS PCSI kit (VSI-I64VMS-WEBUI-V0402-0-1.PCSI or VSI-X86VMS-WEBUI-V0402-0-1.PCSI) that can be installed by a suitably privileged user using the following command:

```
$ PRODUCT INSTALL WEBUI
```

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

```
The following product will be installed to destination:
VSI X86VMS WEBUI V4.2-0                                DISK$THORIN_SYS:[VMS$COMMON.]
```

```
Portion done: 0%...10%...20%...50%...60%...90%...100%
```

```
The following product has been installed:
VSI X86VMS WEBUI V4.2-0                                Layered Product
```

```
VSI X86VMS WEBUI V4.2-0
```

Post-installation tasks are required.

After installation you need to update `civetweb$root:[conf]civetweb.conf` and `civetweb$root:[conf]services.conf` as described in the release notes. After making the necessary changes you can restart the web server using the following command:

```
$ @SYS$STARTUP:WEBUI$RESTART.COM
```

6. Post-Installation Steps

After the installation has successfully completed, the following steps should be performed to configure and start using the WebUI software:

1. Update the system start-up procedure (optional)

After the installation has successfully completed, include the command displayed at the end of the installation procedure into `SYSTARTUP_VMS.COM` to ensure that WebUI is started when OpenVMS is booted.

2. Add identifiers to the rights database

To be able to modify user records, the `SYSUAF` logical name must be defined with a path to the `SYS$UAF.DAT` authorization file. If you have not previously done so on your system, define the logical name as follows before continuing:

```
$ DEFINE/SYSTEM SYSUAF SYS$SYSTEM:SYS$UAF.DAT
```

Access to WebUI functionality is controlled by OpenVMS rights identifiers. Users login to the WebUI using their standard OpenVMS username and password details; however in order to use the WebUI they must also have either the `WEBUI_READ` or the `WEBUI_WRITE` rights identifier, or they may be assigned both identifiers, in which case `WEBUI_WRITE` will take precedence. These identifiers must be added to the rights database before the WebUI can be used (note that the values assigned to the identifiers may differ in your environment; however the values of the identifiers are not important):

```
$ MCR AUTHORIZE
UAF> ADD/IDENTIFIER WEBUI_READ
%UAF-I-RDBADDMSG, identifier WEBUI_READ value %X8001121E added to rights
database
UAF> ADD/IDENTIFIER WEBUI_WRITE
%UAF-I-RDBADDMSG, identifier WEBUI_WRITE value %X8001121F added to
rights database
```

3. Grant rights identifiers to users

Once the rights identifiers `WEBUI_READ` and `WEBUI_WRITE` have been added to the rights database, the identifiers should be granted as appropriate to any users that will be using the WebUI. For example, if user `BIGGLES` requires read and write access to WebUI functions, you would grant them the `WEBUI_WRITE` identifier as follows:

```
UAF> GRANT/IDENTIFIER WEBUI_WRITE BIGGLES
%UAF-I-GRANTMSG, identifier WEBUI_WRITE granted to BIGGLES
UAF>
```

4. Configure the web server

The WebUI installation procedure installs the following sample and template configuration files into the CIVETWEB\$ROOT:[CONF] directory:

Filename	Purpose
CIVETWEB^.CONF.WEBUI	Basic CivetWeb configuration for WebUI
SERVICES^.CONF.WEBUI	Basic CivetWeb services configuration for WebUI
ALERT_IMAGES^.CONF.TEMPLATE	Basic configuration for altering subsystem
THREADS^.CONF.TEMPLATE	Additional images to be loaded and run (as threads) by CivetWeb
WEBUI^.CONF.TEMPLATE	Basic WebUI configuration

If you have not modified the CivetWeb configuration for any other purpose and/or you do not have a previous installation of WebUI, then these configuration files can simply be copied as follows in order to establish an initial working WebUI configuration: ¹

```
$ SET DEFAULT CIVETWEB$ROOT:[CONF]
$ COPY CIVETWEB^.CONF.WEBUI CIVETWEB.CONF
$ COPY SERVICES^.CONF.WEBUI SERVICES.CONF
$ COPY ALERT_IMAGES^.CONF.TEMPLATE ALERT_IMAGES.CONF
$ COPY THREADS^.CONF.TEMPLATE THREADS.CONF
$ COPY WEBUI^.CONF.TEMPLATE WEBUI.CONF
```

If you have made configuration changes for CivetWeb and do not wish to lose them, then it may be necessary to merge the WebUI-specific details with your existing configuration.

Specifically, the `url_rewrite_patterns` configuration setting must be defined as follows in `CIVETWEB.CONF` to ensure that the WebUI REST API operates correctly:

```
url_rewrite_patterns /api/**=/civetweb$root/htdocs/api/api.lua
```

Additionally, the following entries must be included in `SERVICES.CONF` to ensure that WebUI user authentication and authorization work correctly:

```
/login /civetweb$root/local/login.exe LoginHandler
/api/token /civetweb$root/local/login.exe ApiLoginHandler
```

It should be noted that the WebUI installation assumes that the document root for the web server is `/civetweb$root/htdocs`. If this is not the case for your configuration, then additional work will be required to ensure that the WebUI components are served correctly. VSI strongly recommends that you do not change the document root.

Note

If you intend to run WebUI on multiple nodes within the same OpenVMS cluster, it is important to use the same port number for the web server on each cluster node. If different nodes use different port numbers, then the cluster-related functions of WebUI will not operate correctly.

¹Before attempting to copy the configurations files, ensure that extended symbols are enabled on your system. If you have not previously done so, then this can be enabled with the following command:

```
$ SET PROCESS/PARSE=EXTENDED
```

In order to finish the configuration of the embedded terminal, generate a "server.pem" certificate (follow the instructions available at <https://github.com/civetweb/civetweb/blob/master/docs/OpenSSL.md#creating-a-self-signed-certificate>). You must then place this certificate into the CIVETWEB\$ROOT:[RESOURCES.CERT] directory.

5. Define logical names for reports

The reports feature of the WebUI allows you to create command procedures that generate HTML or textual output and make them known to the WebUI so that they can be run by users as and when required to gather various data that might be of interest with regard to the operation and configuration of the OpenVMS system. This reporting facility relies on the definition of the logical names WEBUI\$REPORT_SCRIPTS and WEBUI\$REPORT_FILES. Generally, these logical names would be defined in the system logical name table. However, this is not a strict requirement and they may be defined in other logical name tables so long as they are visible to the CivetWeb web server process.

The logical name WEBUI\$REPORT_SCRIPTS defines the location of any script files (DCL command procedures) and the logical name WEBUI\$REPORT_FILES specifies the location of report files create by the script. Note that it is possible to specify one or more directory paths when defining these logical names and all paths will be searched and displayed by the WebUI when listing available scripts and generated report files.

The WebUI installation includes a sample DCL report script that gathers considerable information about the configuration of the system. This script and its output can be executed and viewed via the WebUI by defining the WEBUI\$REPORT_SCRIPTS and WEBUI\$REPORT_FILES logical names as follows (you may wish to specify alternative locations that are more appropriate to your particular environment):

```
$ DEFINE/SYS WEBUI$REPORT_FILES CIVETWEB$ROOT:[HTDOCS.REPORTS.FILES]
$ DEFINE/SYS WEBUI$REPORT_SCRIPTS CIVETWEB$ROOT:[HTDOCS.REPORTS.SCRIPTS]
```

You should be sure to include these logical name definitions in your system start-up procedure to ensure that they are correctly defined whenever the system is rebooted.

Note

It is possible to specify multiple directories by defining the logical names as a list of directories.

6. SSL configuration (recommended)

The default CivetWeb installation provides only HTTP. If you wish to use HTTPS for the WebUI (recommended), then this will need to be configured. A detailed description of how to configure SSL/TLS and HTTPS is beyond the scope of this document; refer to the open-source CivetWeb documentation.

7. Set the port for WebSockets terminal interface (optional)

In the WEBUI.CONF file (located in the CIVETWEB\$ROOT:[CONF] directory), set the value of `terminal_listening_ports` to the port that you will run WebUI WebSockets terminal interface on.

WebUI must be restarted afterwards for changes to take effect (see [below](#)).

8. Restart the web server

Once you are happy with the configuration of CivetWeb you can cycle the web server to pick up the changes as follows:

```
$ @SYS$STARTUP:WEBUI$RESTART.COM
```

Try to access the web server at <http://hostname:8082> to verify that it is operating correctly (replace "hostname" with the name or IP address of the OpenVMS server on which the CivetWeb web server has been installed and change the port number if necessary, as applicable to your installation).

9. Examine REST API documentation

You may also wish to examine the WebUI REST API documentation, which can be found at <http://hostname:8082/webui/docs/api.html>. This document describes in detail the operations that can be performed via the API, request and response formats, status codes, and so on. You can use this information to develop your own tools to automate various administrative tasks on your OpenVMS systems.

10. Configure alerts database (optional)

Alerts such as intrusions and device errors are stored in a database that by default resides in the `CIVETWEB$ROOT:[HTDOCS.API.ALERTS]` directory. The database is generally not large (less than 9000 blocks) and the default location will generally be acceptable. However, an alternative location may be defined by defining the logical name `WEBUI$ALERTS_ROOT` to specify the desired location for the database. In addition, the logical name `WEBUI$ALERTS_KEEP_COUNT` can be used to specify the maximum number of alert records that will be stored in the database. By default this value is 10000, which should be more than adequate for most systems. If the number of alert records exceeds this value, WebUI will purge the oldest alert records to maintain this maximum value. These two logical names should be defined in the system table to ensure that they will be visible to WebUI.

7. Privileges and Quotas

The WebUI runs inside of the CivetWeb web server and therefore utilises the privileges and quotas that are associated with the username under which CivetWeb is run, which is generally the `SYSTEM` account. It is important to appreciate that many operations that can be performed via the WebUI require a high level of privilege, and care should be taken to ensure that access to WebUI functions is granted only to those users that need it and that those users have only the permissions they require, as determined by the `WEBUI_READ` and `WEBUI_WRITE` rights identifiers.

The WebUI will not typically be expected to support a large number of concurrent users and accordingly there are no specific quota settings required over and above those recommended for CivetWeb itself.

8. Installing in an Alternative Location

It should be noted that because the WebUI runs under CivetWeb it is not possible to use the `PCSI /DESTINATION` qualifier when installing the WebUI to specify an alternative location for the installation. The installation expects the `CIVETWEB$ROOT` logical name to be defined, and installs all files under this location. It is however possible to use the `/DESTINATION` qualifier when installing CivetWeb, and thereby install WebUI in an alternative location.

9. Python Language Bindings

In addition to the web browser-based user interface, WebUI provides a set of Python language bindings for the WebUI REST API that can be used to implement management tools and scripts to perform automated remote management tasks. The Python interface is not provided with the OpenVMS installation kit, but is instead freely available at <https://github.com/vmssoftware/pywebui>. Here you will also find documentation on the API and a number of example Python programs.

10. Known Problems and Limitations

The following known issues and limitations will be resolved in a future release. VSI would welcome information on any bugs that you might find, as well as general feedback regarding how the WebUI product might be improved and enhanced.

- Ensure that there are no existing connections to CivetWeb before stopping and re-starting WebUI. If there are existing active sessions, this can sometimes cause CivetWeb to subsequently crash. To ensure that there are no existing active sessions, users should explicitly log out of WebUI. This issue will be resolved in a future release of the WebUI product.
- License status may not be returned correctly.
- Changes to the license database will be attributed to the username under which CivetWeb is running, as opposed to the user who made the change.
- The Internet Explorer browser and browsers on mobile devices are not currently fully supported. Recommended browsers are Firefox, Chrome, and Safari.
- Via WebUI it is possible to modify various TCP/IP sub-system parameters (VSI and HPE TCP/IP Services only). It should be noted that such changes modify the in-memory configuration only, and changes will not persist upon TCP/IP or system restart.
- For WebUI, it is currently necessary to run CivetWeb under the SYSTEM account. Future releases will provide the ability to use a less-privileged account.