

WebUI 4.1-1 for VSI OpenVMS IA64 and x86-64

June 2022

VSI-I64VMS-WEBUI-V0401-1-1.PCSI
VSI-X86VMS-WEBUI-V0401-1-1.PCSI

[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.

Introduction

Thank you for your interest in the WebUI for VSI OpenVMS IA64 and x86-64. The current release of WebUI for VSI OpenVMS is 4.1-1.

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.

What's new in this release?

This release includes a new module to examine OpenVMS error log messages, in addition to including usability updates and bug fixes to existing modules.

It is also now possible to register licenses (PAKs) by copying and pasting license registration command procedures provided by VSI into a text box provided by the WebUI browser interface. It should be noted that pasted content is very strictly validated such that it is not possible to enter and submit to the server an arbitrary sequence DCL commands via this mechanism.

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 say for sure that you will be so lucky if your system is running older versions.

- VSI OpenVMS 8.4-1H1 or higher (IA64) or VSI OpenVMS Version V9.2 or higher (x86-64)
- VSI TCP/IP or HPE TCP/IP Services for OpenVMS
- CivetWeb 1.14-0C for VSI OpenVMS or higher
- Lua 5.3-5D for VSI OpenVMS or higher (required by CivetWeb)
- PERFDAT 4.8-2 or higher (optional)

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.

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.

Installing the kit

The kit is provided as an OpenVMS PCSI kit (`VSI-I64VMS-WEBUI-V0401-1-1.PCSI` or `VSI-X86VMS-WEBUI-V0401-1-1.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 has been selected:

```
VSI I64VMS WEBUI V4.1-1          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 WEBUI V4.1-1

VMS Software Inc. &

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:

```
VSI I64VMS WEBUI V4.1-1          DISK$IA18_2L3:[VMS$COMMON.]
```

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

The following product has been installed:

```
VSI I64VMS WEBUI V4.1-1          Layered Product
```

VSI I64VMS WEBUI V4.1-1

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
```

Post-installation steps

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

- Update system start-up procedure

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.

- Add identifiers to the rights database

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
```

- 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>
```

- Configure the web server

The WebUI installation procedure installs the following sample and template configuration files into `civetweb$root:[conf]`:

Filename	Purpose
<code>civetweb^.conf.webui</code>	Basic CivetWeb configuration for WebUI
<code>services^.conf.webui</code>	Basic CivetWeb services configuration for WebUI
<code>alert_images^.conf.template</code>	Basic configuration for altering subsystem
<code>threads^.conf.template</code>	Additional images to be loaded and run (as threads) by CivetWeb
<code>webui^.conf.template</code>	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:

```
$ 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
```

If you intend to use the WebUI in conjunction with PERFDAT to monitor and display OpenVMS system performance data then the following line must also be included in `services.conf` and the file `civetweb$root:[conf]perfdat.conf` must be updated as described in the next point.

```
/perfdat /civetweb$root/local/perfdat.exe PerfdatHandler
```

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 the cluster-related functions of WebUI will not operate correctly.

- Integration with PERFDAT (optional)

If you have PERFDAT installed and wish to utilize the WebUI PERFDAT integration capabilities to display graphs and examine system performance metrics within WebUI then it is necessary to modify the file `civetweb$root:[conf]perfdat.conf` to specify the account that should be used to connect with PERFDAT and the node and port number of the PERFDAT gateway. For example, the following would be a typical `perfdat.conf` configuration file, with the PERFDAT gateway listening on port 5254 on localhost and the WebUI connecting to PERFDAT with username BIGGLES.

```
host 127.0.0.1
port 5254
user biggles
pass biggles
debug 1
```

The specified username must exist and should have either `BYPASS` or `OPER` privilege in addition to `NETMBX` and `TMPMBX`; and interactive logins may be disabled (highly recommended). Note that it is possible to specify a node name or TCP/IP address other than that of the node on which the WebUI is installed; however this is generally not recommended as it may cause confusion. The `debug` parameter may be 0 or 1 and controls whether additional information regarding operation of the PERFDAT interface is written to the CivetWeb log file. In general the level of additional information is minimal and VSI recommends that you leave this value set to 1 (which enables additional logging).

- 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 created 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; however 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 that it is possible to specify multiple directories by defining the logical names as a list of directories.](#)

- 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 and you should refer to the CivetWeb documentation at <https://github.com/civetweb/civetweb/blob/master/docs/OpenSSL.md>. Note that this document refers to the location of the SSL certificate file must be set. You will need to specify the path using Linux-style syntax.

- 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
```

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

- 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.

- Optionally configure alerts database

Alerts such as intrusions and device errors are stored in a database that by default resides in the directory `civetweb$root:[htdocs.api.alerts]`. 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.

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 (see above for details of how to run WebUI under an alternative 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.

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.

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.

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 Microsoft Edge and Internet Explorer browsers 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.
- It should be noted that the WebSockets terminal interface uses port 8888. If there is an existing service that uses this port, the terminal interface will fail to start. The port number will be made configurable in a future release of WebUI.