WebUI 3.0-1A for VSI OpenVMS I64

August 2020

Introduction

Thank you for your interest in the WebUI for VSI OpenVMS I64. The current release of WebUI for VSI OpenVMS is 3.0-1A.

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 systems information
- Monitoring of CPU and memory usage
- License management
- Management of system parameters
- Installed products (PCSI only)
- User-defined reports

In addition, the 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.

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 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 Version 8.4-1H1 or higher
- VSI TCP/IP, HPE TCP/IP Services for OpenVMS, or MultiNet TCP/IP
- CivetWeb 1.12.0 for VSI OpenVMS or higher

- Lua 5.3-5A 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.

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-V0300-1A-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 V3.0-1A               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 V3.0-1A

    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 V3.0-1A               DISK$IA21_842L1:[VMS$COMMON.]

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

The following product has been installed:
    VSI I64VMS WEBUI V3.0-1A               Layered Product

VSI I64VMS WEBUI V3.0-1A

    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.

- 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 installation process installs the sample configuration files `civetweb^.conf.webui` and `services^.conf.webui` in `civetweb$root:[conf]`. If you have not modified the CivetWeb configuration for any other purpose then these configuration files can be copied or renamed to `civetweb.conf` and `services.conf` respectively. If you have made configuration changes and do not wish to lose them then it will 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 two 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 issues 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.

- 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 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; 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.

- 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.

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.

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.

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

The following known issues 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 CivitWeb before stopping and re-starting WebUI. If there are existing active sessions this can sometimes cause CivitWeb 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