

# DNP3 MESA-ESS Virtual-Utility Application User Guide

## Home Page

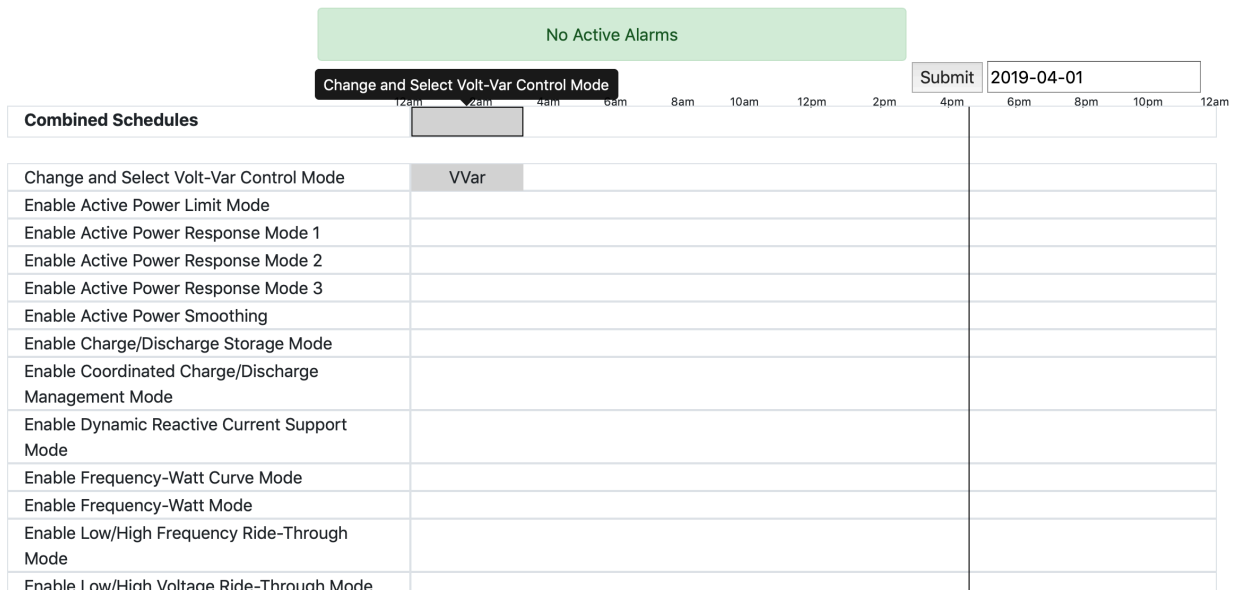
### Notifications

Underneath the main banner, there are a number of notification banners that may appear. Any connection related issues will appear in yellow.

If there are no active alarm points, there will be a green banner indicating that there are 'No Active Alarms.' If any alarms are active, they will be listed in red. Clicking on any active alarms will navigate the user to the **Alarms** page.

### Schedules

Underneath the Notifications, there is a list of **Active** modes and their schedules. At the top of this list is a **Combined** Schedule, outlining the overlap. Hovering over the Combined Schedules will give a pop-up listing the schedules for that time period. To view **Inactive** schedules, click on the combined schedules listing, which will expand to show all modes.



### Favorite Points

A graph at the bottom of the page will show recent history for any points that have been 'favorited' by a user.

## Config Page

### Modes and Functions

There are two columns in the Home Page that pertain to collections of points: Modes and Functions. The difference between these two columns are that Modes contain an Enable point as part of their definition where functions do not.

In the Modes column, there are three pertinent pieces of data.

1. A toggle switch indicating whether or not the mode is Enabled or Disabled. This toggle switch refers exclusively to the value of the Mode Enable point. Setting a toggle to enable will only be possible for modes that have a defined function test (see point #3). If the toggle is clicked and the send action does not receive a SUCCESS response, the toggle will revert and indicate the most recently **Accepted** mode enable value.
2. Mode Name. Clicking on this mode name will navigate the user to the **Function Instance** page where a user can define concrete instances for a given function or mode.
3. Count of defined instances for the given function or mode.

The Functions column contains the Name and Instance Count, but does not have a toggle switch.

### **Types and Categories**

For filtering and categorization of individual points, there are the point type and point category columns. Every defined point in the VUA has a point type. (The 4 most common of these are Analog Input, Analog Output, Binary Input, and Binary Output) Clicking on any of these point types will bring up the **Points** Page, which provides a listing of all the appropriate points.

Point Categories are a categorization mechanism that allows the VUA operator to define a category for select points for easier identification. Two examples of this are 'alarm' points and 'mode enable' points. Alarm points, when given a non-zero value, are displayed in the alarm notification banner at the top of the **Home** page. Mode Enable points signify whether specific modes are active within MESA.

### **Settings**

To connect to an outstation, the Settings button allows a user to define the location of the both the outstation and HTTP Server that handles communication with the outstation. Clicking **Save** will cause the VUA to reconnect to both the server and the outstation.

No Active Alarms

Modes	Functions	Point Types	Point Categories
<span style="background-color: #20c997; color: white; padding: 2px;">Enabled</span> active_power_limit 1	Schedule 0	Analog Input	alarm
<span style="background-color: #6c757d; color: white; padding: 2px;">Disabled</span> active_power_smoothing 0	cease_to_energize 0	Analog Output	mode_enable
<span style="background-color: #6c757d; color: white; padding: 2px;">Disabled</span> automatic_generation_control 0	curve 1	Binary Input	
<span style="background-color: #6c757d; color: white; padding: 2px;">Disabled</span> charge_discharge 0	disconnect 0	Binary Output	
<span style="background-color: #6c757d; color: white; padding: 2px;">Disabled</span> coordinated_charge_discharge 0	frequency_watt_curve 0	Counter	
<span style="background-color: #6c757d; color: white; padding: 2px;">Disabled</span> dynamic_reactive_current 0	frequency_watt_emergency 0		

[Settings](#)

## Points Page

This lists all points for a given filter, either *Type* or *Category*.

### Name

This lists the point name, along with its group, index, and variation, which are used by MESA to identify the point. Clicking on the point name will bring up a pop-up that will allow a user to send a single point value to the outstation. Clicking on this header will sort the points by point name.

For **Binary** points, any input other than *True* will evaluate to False.

### Description

This point description is pulled from the point definition config that is ingested by the VUA during initial config.

### Last value(s) sent/received

The most recent value either sent or received from the VUA. For Input points, these are values received from the outstation. For Output points, this will display the most recently *attempted* sent value, regardless of whether or not the value was successfully sent. To view the value most recently *Accepted*, check out the **History** page. N/A indicates that this point has never been sent or received.

### Time sent/received









This timestamp corresponds to the last value. For output points, this indicates the most recent *Attempt*, not necessarily the most recent *\*Successful\** attempt. N/A indicates that this point has never been sent or received.

### Status

Like Value and Time, this displays the most recent value associated with the point. For input points, this is 'received.' For output points, this could be a number of statuses. The most important being **SUCCESS**, which indicates a successful transmission of a point to the outstation. N/A indicates that this point has never been sent or received. Clicking on this header will sort the points by status value.

### History

## Point Type: Analog Output

Point Name (group/index/variation)	Description	Last value(s) sent/received	Time sent/received	Status	History
<a href="#">DRCT.VrefOfs (40/0/2)</a>		3	Jan. 11, 2019, 10:33 a.m.	SUCCESS	
<a href="#">DOPM.RvrtTms (40/1/2)</a>		N/A	N/A	N/A	
<a href="#">DCTE.WinTms (40/2/2)</a>		7	Jan. 11, 2019, 10:33 a.m.	DOWNSTREAM_FAIL	
<a href="#">DCTE.RvrtTms (40/3/2)</a>		N/A	N/A	N/A	
<a href="#">DOPM.RvrtTms.1 (40/4/2)</a>		N/A	N/A	N/A	
<a href="#">DOPM.WinTms (40/5/2)</a>		N/A	N/A	N/A	
<a href="#">AO.AO6 (40/6/2)</a>		N/A	N/A	N/A	
<a href="#">AO.AO7 (40/7/2)</a>		N/A	N/A	N/A	

Clicking on this button will navigate the user to the specified Point's **History** page, where they can view all historical values of that point.

### Favorite

The last item on the row is a checkbox. Clicking on the checkbox will 'favorite' a point, causing its history to show up in the 'favorites' graph on the **Home Page**.

## History Page

The history page allows a user to view all the past history of a specific point. The page as a whole is broken into two sections, a list of each point value, and a graph of those values.

### List

On the left side of the screen, there is a table of every value sent (or received) to/from the outstation. Each row of the table has 4 columns: timestamp, value, and status indicators, and a delete button. Clicking on the delete button (it looks like a blue x) will delete that individual row from the table of point history. It will also update the graph on the right so the deleted point no longer appears.

To delete historical rows in bulk, use the 'Delete History Older Than' widget. By default, the timestamp is set to the current time, so clicking the button without changing the timestamp will delete ALL historical records. To delete a subset of records, adjust the timestamp widget, then click the 'Delete History Older Than' button.

### Graph

On the right side of the page, underneath the bulk delete widget, there is a graph of the historical data. This chart only graphs data values whose status is *SUCCESS* (for output points) or *received* (for input points). The graph is limited to the most recent 500 valid data points.

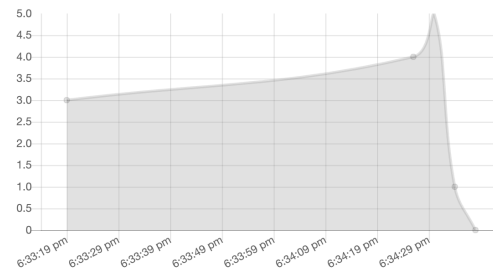
For points with no valid history, or Array Type points, the graph will not display any data.

## DRCT.VrefOfs

Timestamp	Value	Status	
Jan. 11, 2019, 10:33 a.m.	3	SUCCESS	X
Jan. 11, 2019, 10:34 a.m.	4	SUCCESS	X
Jan. 11, 2019, 10:34 a.m.	5	SUCCESS	X
Jan. 11, 2019, 10:34 a.m.	1	SUCCESS	X
Jan. 11, 2019, 10:34 a.m.	0	SUCCESS	X

Delete History Older Than

01/11/2019 10:34



## Function Instance Page

This page is a simple listing of all saved function instances for a given Function or Mode. Clicking on a Function Instance name will navigate a user to the **Function Edit** page. To create a *New Instance*, click on the *Create Function Instance* button. To delete a Function Instance, click on the *Delete* button.

Function Instances for: active\_power\_limit

Create Function Instance

Active Power Limit 1

Delete

## Function Edit Page

### Instance Name, Save, and Save As

All Function Instances must have a name when they are saved. At the top of the Function Edit page, there is an input dialog for the Instance Name. When creating a new Function Instance, this dialog starts out blank, when editing an existing instance, that dialog is pre-populated with the existing Function Instance name.

There are two options for saving a Function Instance: *Save* and *Save As*. When the *Save* button is clicked, the Function Instance will be saved under the name chosen in the dialog at the top of the page. If a user is editing an existing Function Instance, and changed the name of the Instance, that Instance will be renamed accordingly. If a user tries to rename it with the name of a different existing instance, a pop-up will appear, confirming the users decision to overwrite an instance they are not currently editing.

As Opposed to *Save*, *Save As* will create a new copy of a Function Instance. Clicking the *Save As* button will prompt a new pop-up, allowing a user to provide a new name for the copied Function Instance. Similar to the *Save* functionality, if the name provided is already taken by another saved Function Instance, a pop-up confirming the decision will appear.

After both a *Save* or *Save As* command is issued, the VUA will redirect the user to the **Function Instance** page.

## Deleting Function Instance

After a Function Instance has been saved, if a user navigates to edit that instance via the **Function Instance** page, a *Delete* button will appear at the bottom of the page. Clicking on this button will delete the Function Instance and navigate the user back to the **Function Instance** page.

## Upload and Export JSON and CSV Files

Users can upload CSV or JSON files populate Function Instances using the Upload File widget. Use the Choose File button to select the desired CSV or JSON.

To export a Function Instance configuration, click on the *Export* button at the bottom of the page, and then choose whether to download the file as CSV or JSON. After selecting file type, the download should automatically start.

## Function Instance Step Table

The majority of the Function Edit page is taken up with a table outlining all of the steps in that function. Each step corresponds to a point and together they constitute a MESA function or mode.

Columns in the Step Table

1. Step Number
2. Point Name - This is the name of the point being described by the step. Similar to the **Points** page, clicking on the name of this point will bring up a pop-up where a user can send an individual point value to the Outstation.
3. Description - Point description loaded from the point definitions. If there is no description in the point defs, then this field will be empty.
4. Flags - Outlines various flags for the step. The most important of these is the *Optional* flag, which has values of *Optional* or *Mandatory*. To be MESA-compliant, function instances must have values for all *Mandatory* steps when they are sent to the outstation.
5. Value(s) - This is where a user can define values for each step in a Function or Mode. There are three types of input into this column
  1. Float Field - The most common of the three. Blank text field that can contain any float value.
  2. Array Type Field - Some points are Array Type points. These points will have a button next to them titled *Enter array value(s)*. Clicking on that button will open up an Array Field pop-up where a matrix of point values can be defined.
  3. Reference Points - The Final type of point is a reference point. Some mode points contain references to other functions: Curves and Schedules. In places where these references exist, a picklist will appear allowing a user to select which Function they want to relate to the Mode. Next to the picklist, there is also an *Edit* button, which can be used to edit the referenced Function. Depending on the value chosen in the picklist, when the *Edit* button is clicked, the edit popup will either contain the referenced Function, or will be blank if *New* is the selected value, allowing a user to create a totally new Child Function.
6. History - Clicking on this button will navigate the user to the specified Point's **History** page, where they can view all historical values of that point.

## Sending Functions

The final button on the Page is the *Send* button. Clicking the *Send* button will send all of the current values in the Step Table to the Outstation as one Function. When the function has completed sending, a new **Results** page will appear, listing all of the sent points and values, along with the Status returned by the Outstation.



# Mesa Virtual Utility Application

[Home](#)[Logout](#)

## Function: active\_power\_limit

Function Instance Name:

### Upload File

No file chosen

Step	Point Name	Description	Flags	Value(s)	
1	DAMG.ECPRefId		Optional: Mandatory.	<input type="text" value="New"/> <input type="button" value="Edit"/>	<input type="button" value="E"/>
2	DAMG.WRef.out		Optional: Mandatory.	<input type="text" value="4.5"/>	<input type="button" value="E"/>
3	DAMG.WinTms.out		Optional: Optional.	<input type="text"/>	<input type="button" value="E"/>
4	DAMG.RmpTms.out		Optional: Optional.	<input type="text"/>	<input type="button" value="E"/>
5	DAMG.RvrtTms.out		Optional: Optional.	<input type="text"/>	<input type="button" value="E"/>
6	DAMG.WImpLimPct.out		Optional: Optional.	<input type="text"/>	<input type="button" value="E"/>
7	DAMG.WExpLimPct.out		Optional: Optional.	<input type="text"/>	<input type="button" value="E"/>
8	DAMG.ModPrty.out		Optional: Optional.	<input type="text"/>	<input type="button" value="E"/>