# Template Host Type Manual

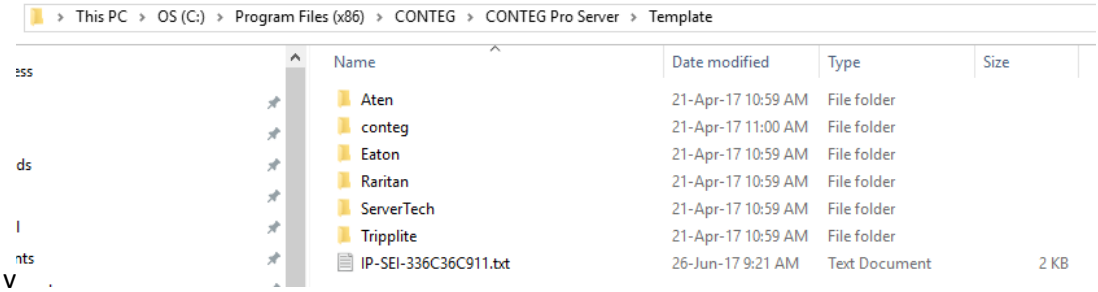
Table of Contents

## Introduction

The *Template Host Type* is a new host type that allows the user to choose a template file that contains a set of pre-set virtual sensor configuration, so they aren't needed to be added one by one. These sensors will be added to the network device host automatically upon adding it to the CPS console.

This feature is useful for customers who would like to monitor a network device by using a variety of virtual sensors.
For example, a customer has their own product that provides sensor data by SNMP GET.
They can use CPS to monitor their product by creating SNMP Virtual Sensors. But for many sensors, they have to recreate a lot of Virtual Sensors for this network device. Instead, they can use this template to automatically ganerate a lot of Virtual Sensors automatically, depending on the configuration file.

The template file has to be stored in the directory named '**Template**' in the server's installed location.
By default it's **C:\Program Files\CONTEG\Conteg Pro Server\Template**




The template file uses the comma-separated values (CSV) file format and the file type must be ".txt".

If the user would like to add lots of hosts at once, it can be done by the *Host Group File* feature with additional configuration. More details can be found in the manual of this feature.

## Template File Configuration Details

**<sensor id of the file>,<virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<analog/switch configuration>,<other configuration>**

| | |
|---|---|
| sensor id of the file | it's an internal reference ID for a sensor in the template host configuration, it doesn't get in conflict with other sensor IDs. It must be unique only in one configuration template. |
| virtual sensor type | "snmp" is Virtual SNMP GET Sensor.<br>"script" is Virtual Custom Script Sensor.<br>"modbus" is Virtual Modbus Sensor.<br>"multiple" is Virtual Multiple Sensor. |
| sensor name | sensor name |
| sensor style | 0 is switch sensor type.<br>1 is analog sensor type.<br>2 is static sensor type. (available only for the custom script sensor) |
| graph enable | 1 is graph enable.<br>0 is graph disable. |
| polling interval | number of seconds. |
| timeout | number of seconds. |
| error retrying | number of times. |
| analog/switch configuration | This will vary, the necessary configuration depends on the virtual sensor style. |
| other configuration | This will vary, the necessary configuration depends on the virtual sensor style. |

**Switch configuration**

<state>,<state value>,<off description>,<on description>

| | |
|---|---|
| state | 0 is normal state<br>1 is critical state |
| state value | This is used for comparison with the sensor value reading; if the sensor value equals with this value, the sensor's status will be normal/critical depending on the set state value:<br><br>if state = 0, the other status is critical<br>if state = 1, the other status is normal<br><br>**For Virtual Multiple Sensors:**<br>0 is Critical when All False.<br>1 is Critical when All True.<br><br>Programmatically it looks like this:<br>if state value == sensor value then<br>  status = critical<br>else<br>  status = normal |

Examples:

sensor status is critical if the reading is anything other than 20:
- state value = 20 and state = 0 (normal)
- sensor value reading = 10 -> sensor state critical
- sensor value reading = 20 -> sensor state normal

sensor status is critical only if the reading matches exactly 30:
- state value = 30 and state = 1 (critical)
- sensor value reading = 10 -> sensor state normal
- sensor value reading = 30 -> sensor state critical

Note: the sensor will be in sensor error only when the script can't be executed to get the sensor value.

**Analog Configuration**

<min>,<max>,<high critical>,<high warning>,<low warning>,<low critical>,<rearm>,<factor>,<unit>

| | |
|---|---|
| factor | Available values are 1,0.1,0.01,0.001,10,100,1000. ( value = raw value * factor ) |

**Static configuration**

No additional configuration is needed.

**Virtual SNMP configuration**

<OID>,<Data Type>,<Separator>,<Index>

| | |
|---|---|
| Data Type | 0 means number.<br>1 means string. |
| Separator | This is used for parsing string data (see below).<br>0 means comma separator.<br>1 means semicolon separator. |
| Index | The index parameter lets you choose the actual sensor reading data by index after parsing the string data from the SNMP GET result. This should only be used when the result is a comma- or semicolon separated string with multiple numbers; if the result is a single number, the index must be 0. |

*Example 1*

* OID .1.2.3.4.5.6
* switch style
* normal state 5
* on description is "Activated"
* off description is "Deactivated"

<sensorid>,<virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off description>,<OID>,<Data Type>,<Separator>,<Index>

0,snmp,Sensor Name,0,0,15,15,3,0,5,Activated,Deactivated,.1.2.3.4.5.6,0,0,0

*Example 2*

* OID .1.2.3.4.5.6
* SNMP result is a string "10,20,30,40,50"
* the actual data is at index 3: in this case it is 30.
* switch style
* critical state is 10
* on description is "Activated"
* off description is "Deactivated"

<sensorid>,<virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off description>,<OID>,<Data Type>,<Separator>,<Index>

0,snmp,Sensor Name,0,0,15,15,3,1,10,Activated,Deactivated,.1.2.3.4.5.6,1,0,3

**Virtual Custom Script configuration**

See below for detailed instructions about how to add a custom script file.

<Script Name>,<Script Parameter>

*Example*
* script name is "random.exe"
* script parameter is "-m 100"
* analog style
* min value is 0
* max value is 100
* status threshold is 20,40,60,80
* rearm is 3
* factor is 10.
* unit is "T"

<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<min>,<max>,<high critical>,<high warning>,<low warning>,<low critical>,<rearm>,<factor>,<unit>,<Script Name>,<Script Parameter>

0,script,Sensor Name,1,0,15,15,3,0,100,80,60,40,20,3,10,T,random.exe,-m 100

**Virtual Modbus configuration**

<MODBUS port>,<MODBUS command>, <MODBUS data order>,<MODBUS data type>,<MODBUS data factor>,<MODBUS address>,<MODBUS slave id>

| | |
|---|---|
| MODBUS command | 1 is read coil status<br>2 is read input status<br>3 is read holding registers<br>4 is read input registers |
| MODBUS data order | 1 is Low Byte First, Low Word First<br>2 is Low Byte First, High Word First<br>3 is High Byte First, High Word First<br>4 is High Byte First, Low Word First |
| MODBUS data type | 1 is 16 bits unsigned int<br>2 is 16 bits signed int<br>3 is 16 bits two characters ASCII<br>4 is 32 bits unsigned int<br>5 is 32 bits signed int<br>6 is 32 bits IEEE floating point |
| MODBUS data factor | This factor will apply only for commands "read holding registers", "read input registers"<br>( value = raw * factor ) |
| MODBUS address | 0-65535 |
| MODBUS slave id | 0-255, default is 255 |

*Example*

* MODBUS port is 502
* MODBUS command is 4 (read input registers)
* MODBUS data order is 1 (Low Byte First, Low Word First)
* MODBUS data type is 5 (32 bits signed int)
* MODBUS data factor is 0 (The value will not be applied by this factor because data type is 32 bits signed int)
* MODBUS address is 16 (0x10)
* switch style
* normal state 0
* on description is "Present"
* off description is "Not Present"
* MODBUS slave id is 255 (default value)

<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off

```
description>,<MODBUS port>,<MODBUS command>, <MODBUS data order>,<MODBUS data type>,<MODBUS
data factor>,<MODBUS address>,<MODBUS slave id>

0,modbus,Sensor Name,0,0,15,15,3,0,0,Present,Not Present,502,4,1,5,0,16,255
```

**Virtual Multiple sensors configuration**

<calculation>,<sensor-status pairs>

| | |
|---|---|
| calculation | 0 is a summary (addition) of all sensors in sensor-status pairs. Ex. 2+3+1=6<br>1 is division. (1st sensor / 2nd sensor).<br>Note: If division by zero occurs, then the sensor's status will become "sensor error". |
| sensor-status pairs | The format is <sensor-id;status;sensor-id;status> up to a maximum of 8 pairs.<br><br>possible values for status:<br>no status = 1<br>normal = 2<br>high warning = 3<br>high critical = 4<br>low warning = 5<br>low critical = 6<br>error = 7<br>low out = 8<br>high out = 9<br>no voltage present = 10<br>voltage present = 11<br>unreachable = 15 |

*Example 1 (switch style with summary)*
* sensor-pair:
  id 10 status normal 2
  id 11 status high critical 3
* switch style
* Critical when all true
* on description is "Critical"
* off description is "Normal"

<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off description>,<calculation>,<sensor-status pairs>

0,multiple,Sensor Name,0,0,15,15,3,0,0,Critical,Normal,0,10;2;11;3

*Example 2 (analog style with summary)*
* sensor-pair to summary:
  id 10 status not used

```
  id 11 status not used
* analog style
* min value is 0
* max value is 100
* status threshold is 20,40,60,80
* rearm is 3
* factor is 10.
* unit is "T"

<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling
interval>,<timeout>,<error retrying>,<min>,<max>,<high critical>,<high warning>,<low
warning>,<low critical>,<rearm>,<factor>,<unit>,<calculation>,<sensor-status pairs>

0,multiple,Sensor Name,1,0,15,15,3,0,100,80,60,40,20,3,10,T,0,10;0;11;0
```

**All format templates**

**SNMP sensor - switch style**
<sensorid>,<virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off description>,<OID>,<Data Type>,<Separator>,<Index>

**SNMP sensor - analog style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<min>,<max>,<high critical>,<high warning>,<low warning>,<low critical>,<rearm>,<factor>,<unit>,<OID>,<Data Type>,<Separator>,<Index>

**Custom Script sensor - switch style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off description>,<Script Name>,<Script Parameter>

**Custom Script sensor - analog style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<min>,<max>,<high critical>,<high warning>,<low warning>,<low critical>,<rearm>,<factor>,<unit>,<Script Name>,<Script Parameter>

**Custom Script sensor - static style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<Script Name>,<Script Parameter>

**Modbus sensor - switch style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off description>,<MODBUS port>,<MODBUS command>, <MODBUS data order>,<MODBUS data type>,<MODBUS data factor>,<MODBUS address>,<MODBUS slave id>

**Modbus sensor - analog style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<min>,<max>,<high critical>,<high warning>,<low warning>,<low critical>,<rearm>,<factor>,<unit>,<MODBUS port>,<MODBUS command>, <MODBUS data order>,<MODBUS data type>,<MODBUS data factor>,<MODBUS address>,<MODBUS slave id>

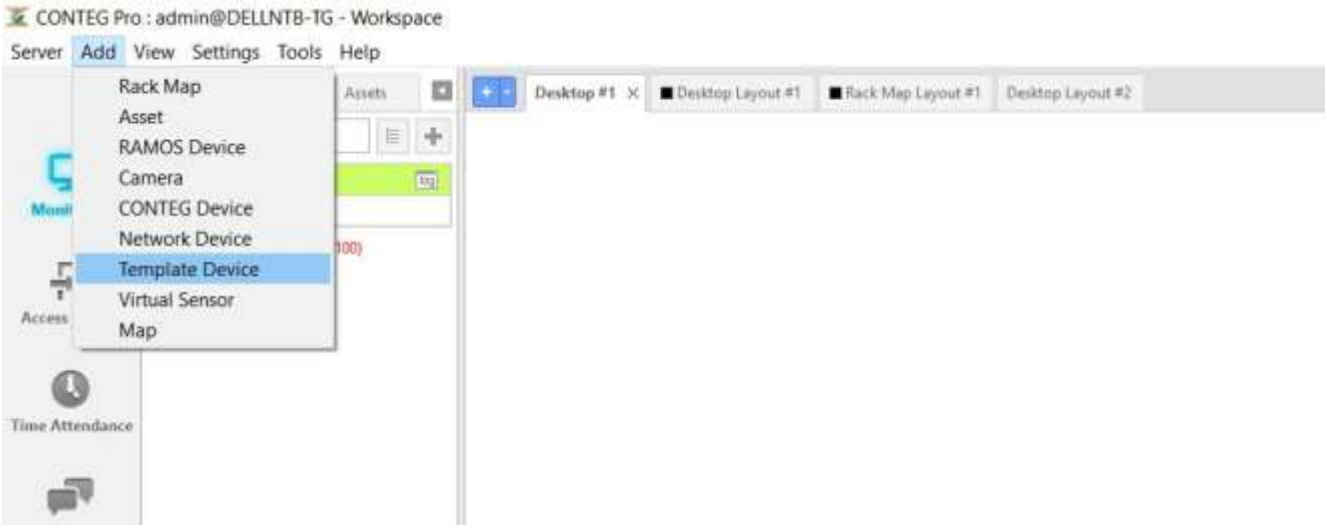
**Multiple sensors - switch style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<state>,<state value>,<on description>,<off description>,<calculation>,<sensor-status pairs>

**Multiple sensors - analog style**
<sensorid><virtual sensor type>,<sensor name>,<sensor style>,<graph enable>,<polling interval>,<timeout>,<error retrying>,<min>,<max>,<high critical>,<high warning>,<low warning>,<low critical>,<rearm>,<factor>,<unit>,<calculation>,<sensor-status pairs>

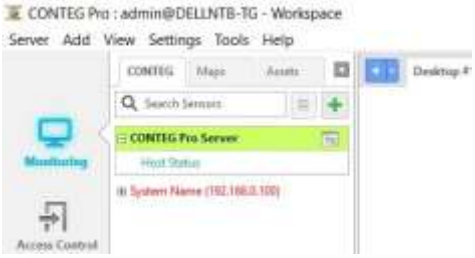
## User Interface Guide

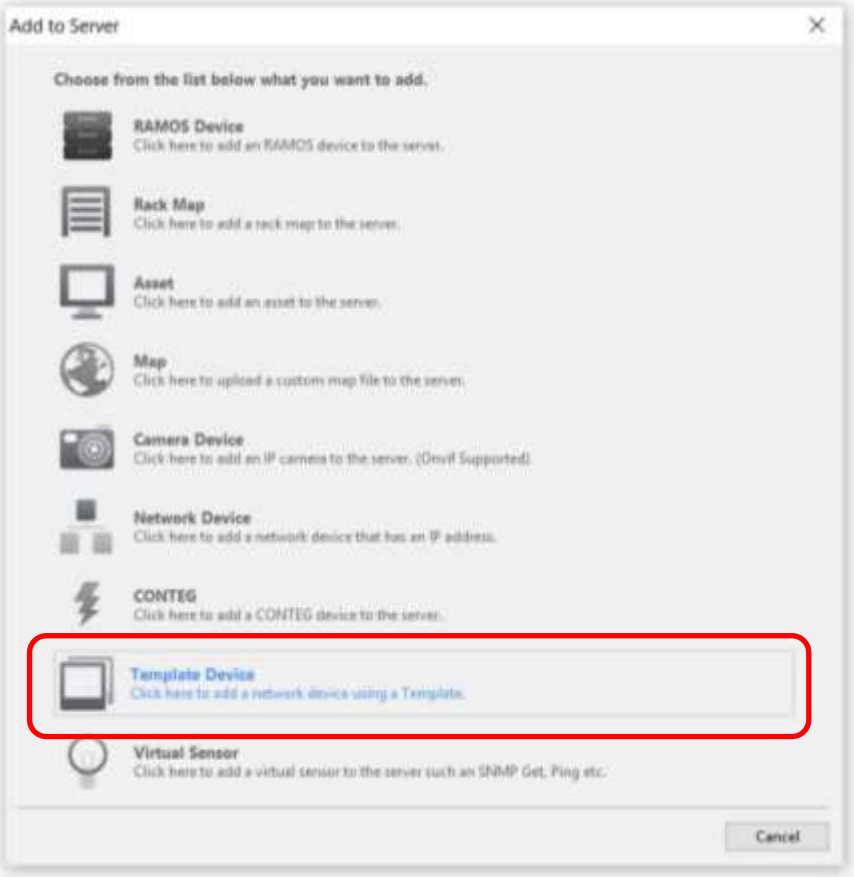
First you'll have to prepare the template file before adding the Template Device to the console.
See the Example Setup section in this manual below for details.

You can add a new Template Device by using the **Add menu / Template Device** or by clicking the "**Add**" menu at the toolbar, then clicking  the "**Template Device**" menu.

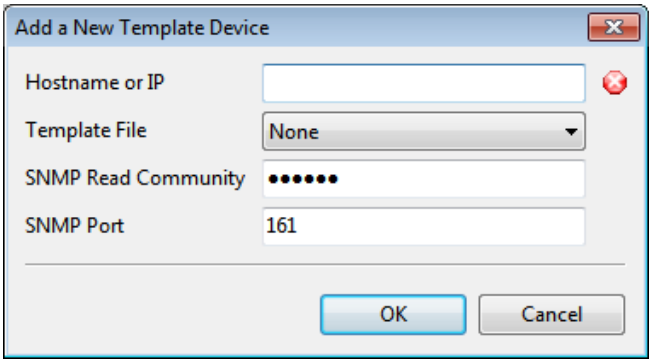



Or use [+] button




Select Template Device from the list.



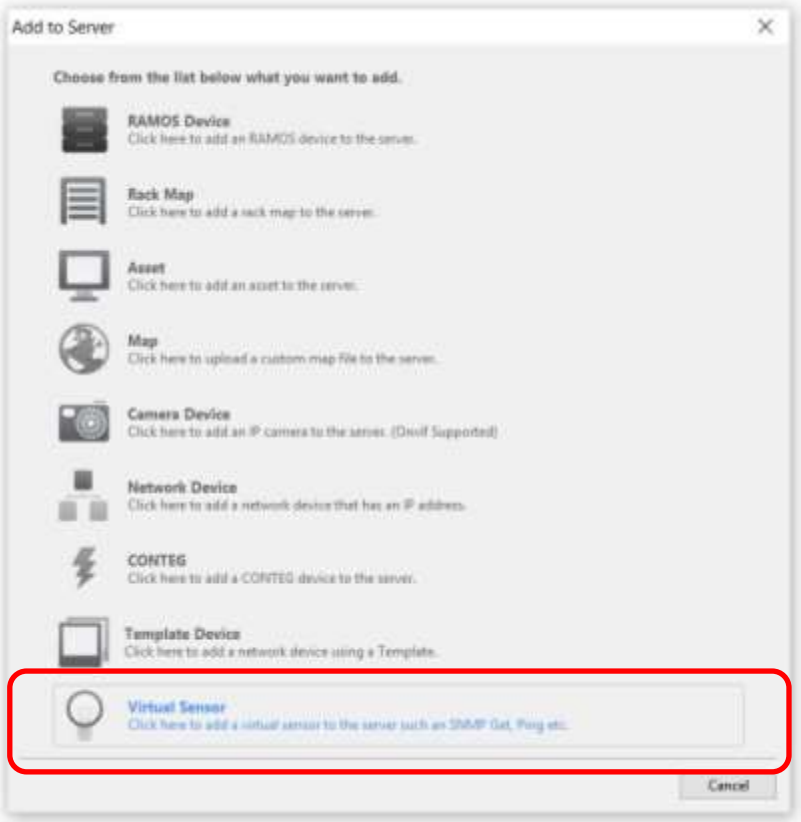

Specify the network device's IP or hostname, and select the previously created template file from the drop-down menu. Specify the SNMP read community and port, if they are different from the default values.

Click OK to finish.

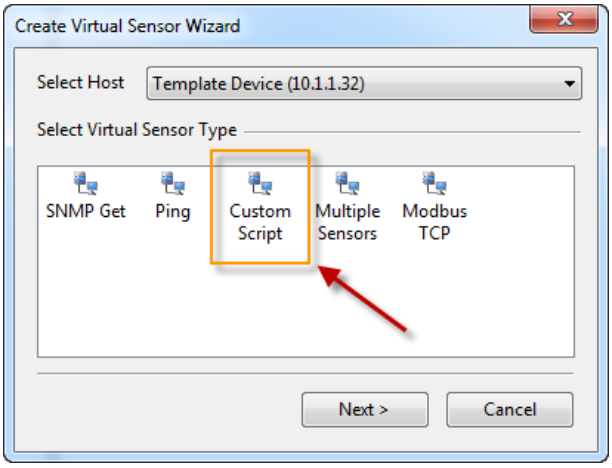
The virtual sensors will be automatically added to the Template Device.

## How to Add a Custom Script File

Click the "**Add**" button at the bottom of the server explorer panel, then click "**Virtual Sensor**"; or click the "**Add**" menu at the toolbar, then click the "**Virtual Sensor**" menu.




Choose the "**Custom Script**" option (it can be done with any host selected).



Click the "**Add Your Own Script**" button. It will show a dialog window.

Browse to your script, then click the "**Add File**" button.


Alternatively, you can copy your script manually into the storage path below:

**C:\ProgramData\CONTEG\Conteg Pro Server\VirtualSensor\Custom**

*Note:*
For previous versions of Windows (XP, 2003) the location is
**C:\Documents and Settings\All Users\CONTEG\Conteg Pro Server\VirtualSensor\Custom**

## Example setup

In this example, we'll create a Template Device with 3 virtual sensors attached:

- Ping host (using custom script pinger.bat)
- Temperature Sensor Status (using SNMP GET)
- Temperature Sensor Value (using SNMP GET)

These will be defined in the Template Device file **test.txt.** Be sure to review each section to get a better understanding of the template file.

### A) Ping sensor setup

Create the batch file here: **C:\ProgramData\CONTEG\Conteg Pro Server\VirtualSensor\Custom**
(or use Add Custom Script from the GUI as described earlier in this manual)

Script contents:

```
@echo off
ping %1 | findstr unreachable >NUL
if %errorlevel% EQU 0 echo 1
if %errorlevel% NEQ 0 echo 0
```

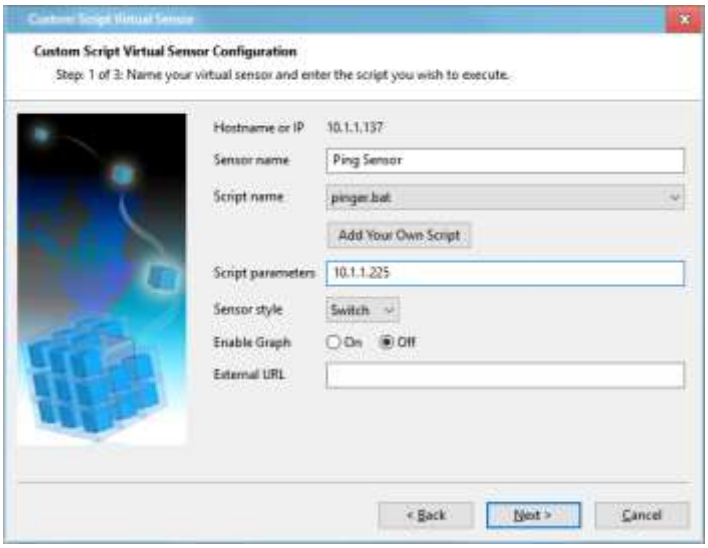
This script file will ping the host specified as a parameter (%1, in our case 10.1.1.225), and will set the end result of the script depending on the ping result. If the host is reachable, it will return 0; if it's unreachable, returns 1.

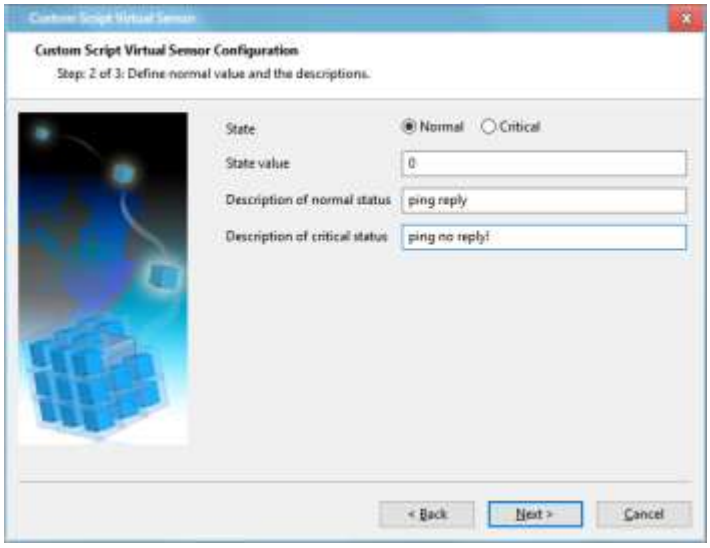
*Important Note:*
The script itself will run on the server machine where CPS is installed and is independent of the Template Host network device where it's attached to in the CPS console. For example this ping script will still run by itself, even if the network device is unreachable.
Also, you'll have to use Windows script commands and programs in the script.

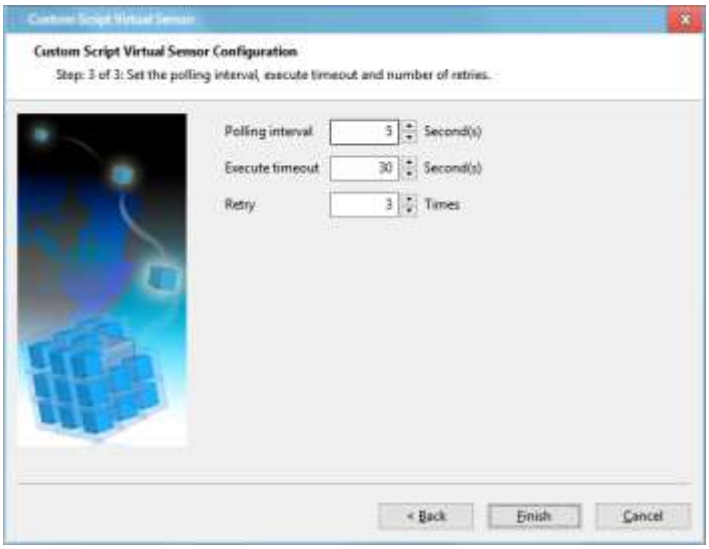
In the template file (test.txt) this line configures the virtual ping sensor:
`0,script,Test Ping Sensor,0,0,5,30,3,0,0,ping no reply!,ping reply,pinger.bat,10.1.1.225`

This sensor has the sensor ID 0, as the first sensor in the list (see the details about sensor IDs earlier in this manual). You need to ensure that the ID is unique, but it only needs to be unique in the actual file.

It is equivalent as the GUI configuration of these settings:






Check the Template File Configuration Details for changing the pre-set parameters in the template file (test.txt).

## B) Temperature Sensor Status setup

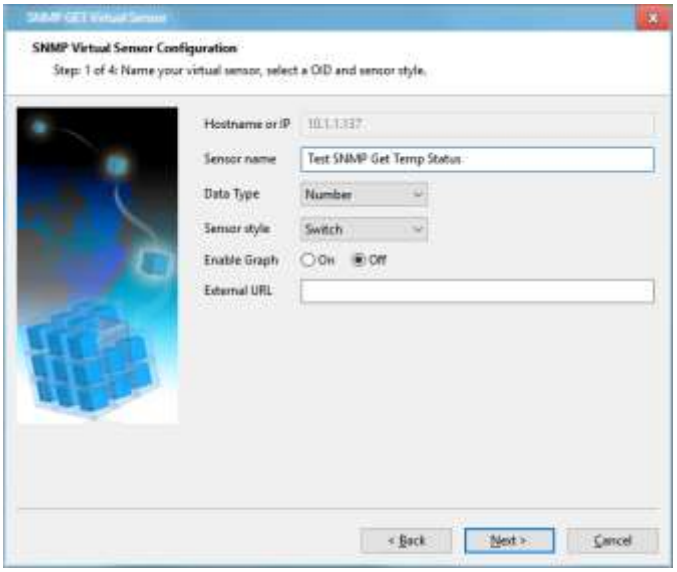
This is a simple SNMP GET type Virtual Sensor, for checking the status of the given sensor (SNMP OID). It's a switch style SNMP sensor. Like the ping sensor, it runs on the CPS machine.

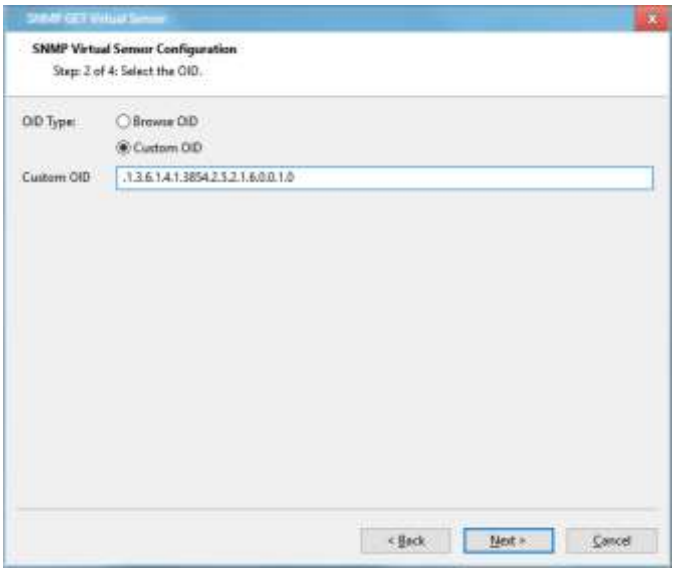
In the template file (test.txt) this line configures the virtual status sensor:

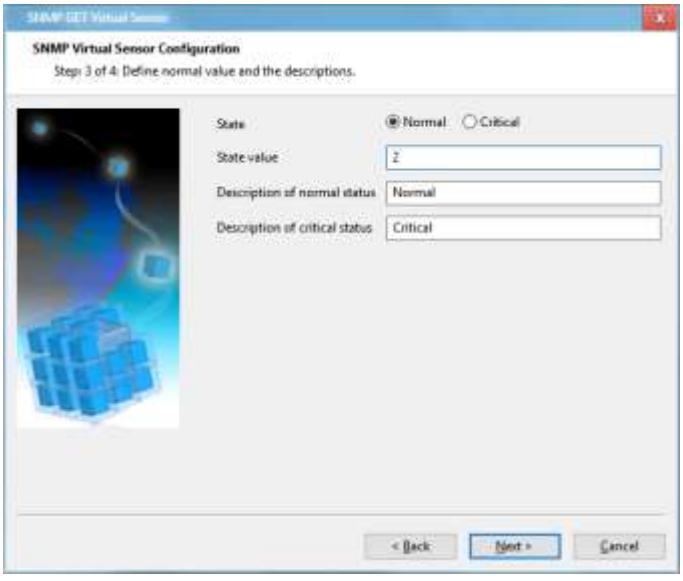
```
1,snmp,Test SNMP GET Temp
Status,0,0,5,10,3,0,2,Critical,Normal,.1.3.6.1.4.1.3854.2.5.2.1.6.0.0.1.0,0,0,0
```

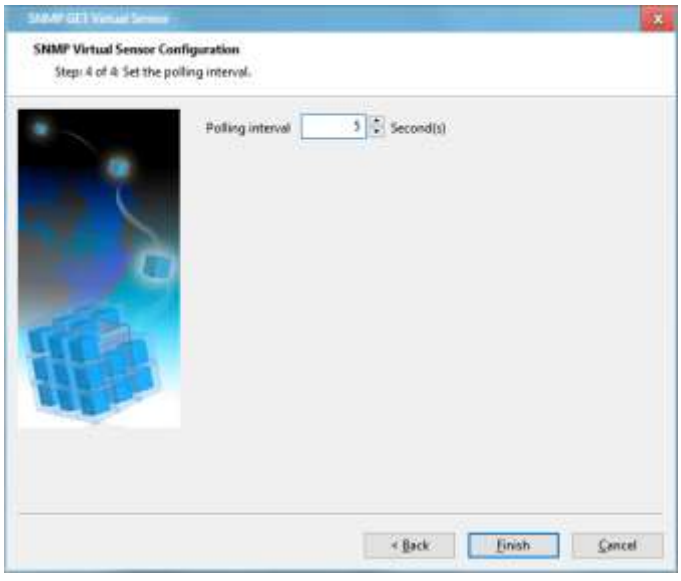
Tip: Get the actual OID values from the Web UI of the unit if available, or use a MIB browser.

It is equivalent as the GUI configuration of these settings:




Check the Template File Configuration Details for changing the pre-set parameters in the template file (test.txt).

## C) Temperature Sensor Value setup

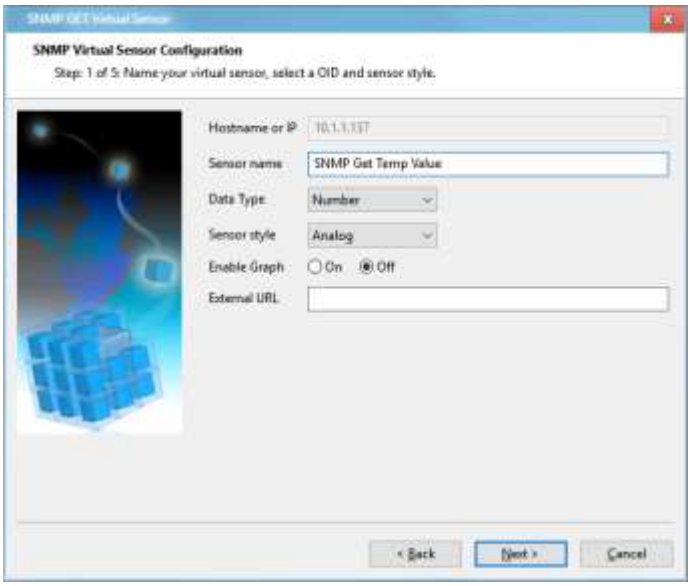
This is a simple SNMP GET type Virtual Sensor, for checking the decimal reading value of the given sensor (SNMP OID). It's an analog style SNMP sensor. Like the ping sensor, it runs on the CPS machine.

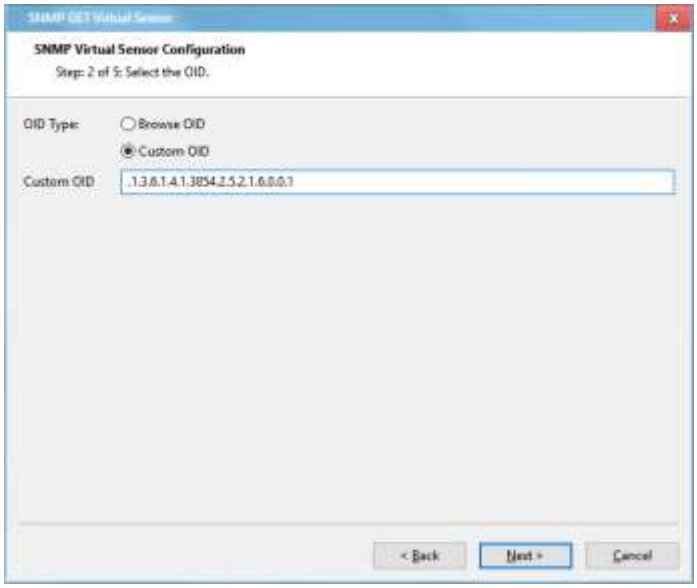
In the template file (test.txt) this line configures the virtual status sensor:

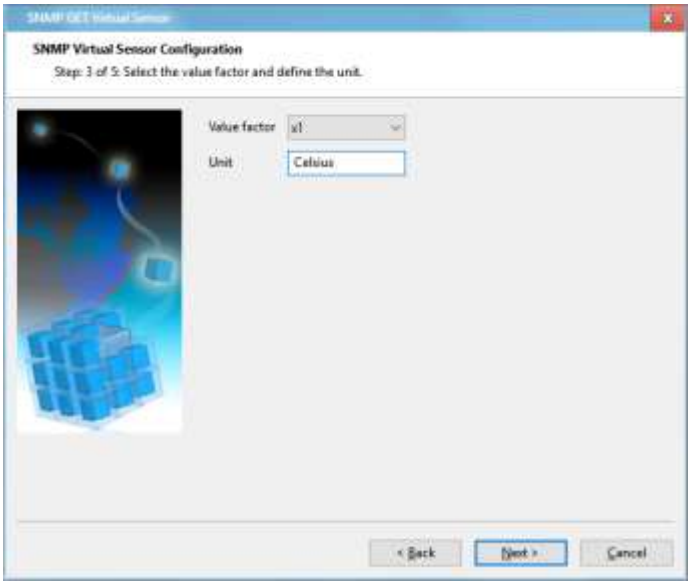
```
2,snmp,Test SNMP GET Temp
Value,1,0,5,10,3,5,50,40,30,20,10,2,1,Celsius,.1.3.6.1.4.1.3854.2.5.2.1.4.0.0.1.0,0,0,0
```

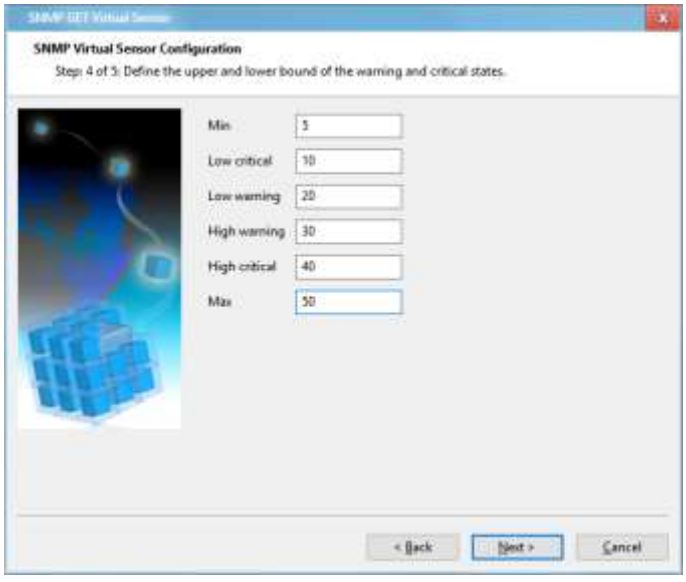
Tip: Get the actual OID values from the Web UI of the unit if available, or use a MIB browser.

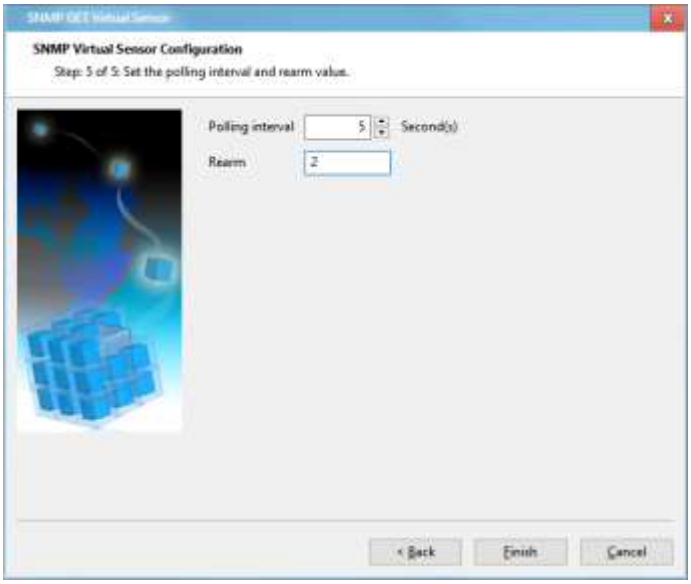
It is equivalent as the GUI configuration of these settings:





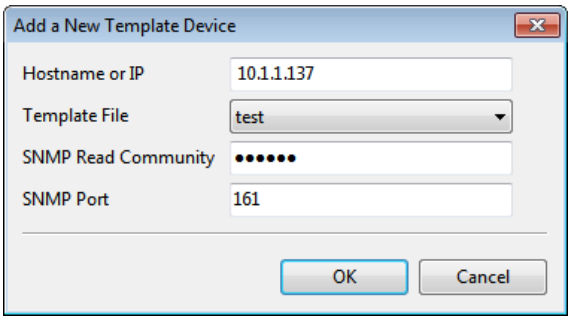

Check the Template File Configuration Details for changing the pre-set parameters in the template file (test.txt).

### Full test template file details (test.txt)

This is the full template file that contains all 3 virtual sensors. It has to be in the Template directory of the Server. The default path is:
**C:\Program Files\CONTEG\Conteg Pro Server\Template**

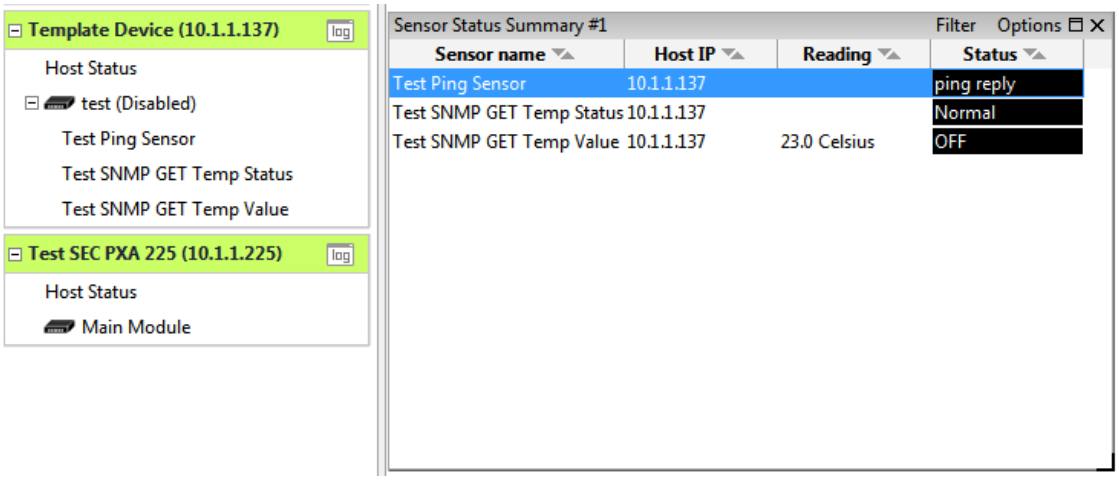
```
0,script,Test Ping Sensor,0,0,5,30,3,0,0,ping no reply!,ping reply,pinger.bat,10.1.1.225
1,snmp,Test SNMP GET Temp
Status,0,0,5,10,3,0,2,Critical,Normal,.1.3.6.1.4.1.3854.2.5.2.1.6.0.0.1.0,0,0,0
2,snmp,Test SNMP GET Temp
Value,1,0,5,10,3,5,50,40,30,20,10,2,1,Celsius,.1.3.6.1.4.1.3854.2.5.2.1.4.0.0.1.0,0,0,0
```

After you've prepared the file, add a Template Device as described in the User Interface Guide:




Specify the network device's IP or hostname, and select the "**test**" Template File from the drop-down list.
Specify the SNMP read community and port, if they are different from the default values.
The virtual sensors will be automatically added to the Template Device.

The new Template Device should look similar to this in the console:




You can modify each virtual sensor's setup later on, as you would do with the manually added ones.

*Note:* You can't delete the predefined sensors individually from the Template Device. You need to delete the Template Device if you want to reapply your configuration.

### Example template file for multiple sensors

```
1,snmp,Temperature,1,1,15,15,3,0,100,35,30,20,10,0,0.1,°C,.1.3.6.1.4.1.17420.1.2.7.0,0,0,0
2,snmp,Humidity,1,1,15,15,3,0,100,80,60,40,20,0,0.1,%,.1.3.6.1.4.1.17420.1.2.8.0,0,0,0
3,snmp,Current A,1,1,15,15,3,0,16,14,12,2,1,0,0.1,A,.1.3.6.1.4.1.17420.1.2.6.0,1,0,1
4,snmp,Current B,1,1,15,15,3,0,16,14,12,2,1,0,0.1,A,.1.3.6.1.4.1.17420.1.2.6.0,1,0,2
5,snmp,Current C,1,1,15,15,3,0,16,14,12,2,1,0,0.1,A,.1.3.6.1.4.1.17420.1.2.6.0,1,0,3
6,snmp,Circuit Status A,0,0,15,15,3,1,-2,Tripped,Normal,.1.3.6.1.4.1.17420.1.2.6.0,1,0,1
7,snmp,Circuit Status B,0,0,15,15,3,1,-2,Tripped,Normal,.1.3.6.1.4.1.17420.1.2.6.0,1,0,2
8,snmp,Circuit Status C,0,0,15,15,3,1,-2,Tripped,Normal,.1.3.6.1.4.1.17420.1.2.6.0,1,0,3
9,snmp,Net Used Status A,0,0,15,15,3,1,-1,Net Used,Normal,.1.3.6.1.4.1.17420.1.2.6.0,1,0,1
10,snmp,Net Used Status B,0,0,15,15,3,1,-1,Net Used,Normal,.1.3.6.1.4.1.17420.1.2.6.0,1,0,2
11,snmp,Net Used Status C,0,0,15,15,3,1,-1,Net Used,Normal,.1.3.6.1.4.1.17420.1.2.6.0,1,0,3
12,multiple,Total Current,1,1,15,15,3,0,48,42,36,6,3,0,1,A,0,3;0;4;0;5;0
```

```
1,snmp,Test Get Value A,1,1,15,15,3,0,16,14,12,2,1,0,0.1,Unit,.1.3.6.1.4.1.17420.1.2.6.0,1,0,1
2,snmp,Test Get Value B,1,1,15,15,3,0,16,14,12,2,1,0,0.1,Unit,.1.3.6.1.4.1.17420.1.2.6.0,1,0,2
5,snmp,Test Get Value C,1,1,15,15,3,0,16,14,12,2,1,0,0.1,Unit,.1.3.6.1.4.1.17420.1.2.6.0,1,0,3
10,multiple,Total Value,1,1,15,15,3,0,48,42,36,6,3,0,1,Unit,0,5;0;2;0;1;0
```

This picture shows an example for multiple sensor configurations (sensor ID 12).
The referenced sensor IDs are highlighted to show their position in the multiple sensors line.

We won't go into detail about configuring the multiple sensors in this manual, but you can find the required parameters for the configuration in the first section.

**Please contact support@conteg.com if you have any further technical questions or problems.**

## Thanks for Choosing Conteg Pro!