

- Copy and paste the content of Template in the target folder.

If in Toolbox, delete template/help.htm and rename helpToolbox.htm as help.htm

If in InfoScript, delete helpToolbox.htm

Rename the files and folder

-Rename the folder template, and the _d, _i, _m, _p script file with the device type devType of the device

Adapt the method names and path in script

- In _d, _i, _m, _p script, replace anytime template is written by the device type of the device

Adapt the HTML script path

-In devType folder (previously named template), adjust the path of the htm file according to the new location.

Adapt the HTML files location path

-in _m, in the method devType _open_o (previously named template_open_o), in line 24, adapt the path in order to locate the hmtl.

f provides the path of EMTPWorks: ex: C:\Program Files (x86)\EMTPWorks 3.4_dev\

Call i when double clicking on the device in EMTP

- In EMTP, right click to the device intended to open the mask and go subcircuit info/Mask/Mask this subnetwork/Specify a script to use to display properties. Browse... and select the _i script file.

Verification:

At this point, when double clicking on the mask, the following mask should open:

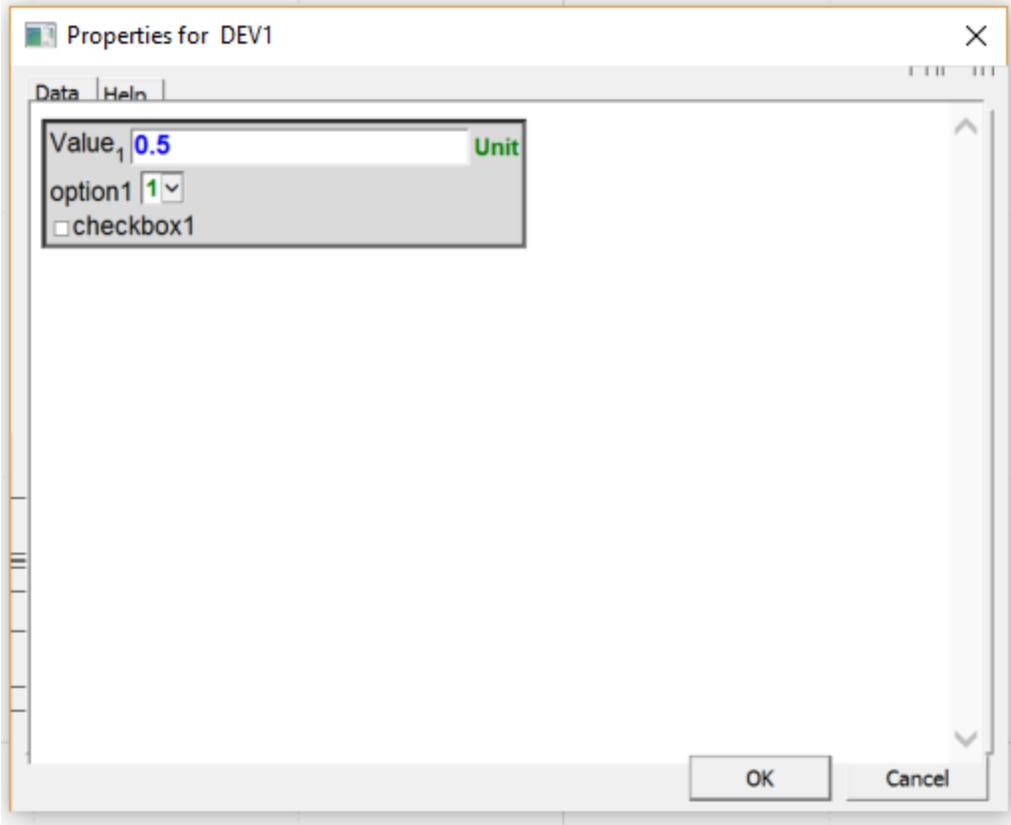


Figure 1: template mask

Define the parameters whose value are defined in the mask

- Go to the `_m` file, in the function `oDevice_devType_define_me()` (previously named `oDevice_template_define_me`), enter all the parameters that have to be defined in the different tabs of the mask and assigned them a default value. The default value is given when the attribute `FormData` is empty.

Build the HTML

- Build the HTML using the parameter defined in `oDevice_devType_define_me()` (previously named `oDevice_template_define_me`)
- The HTML files are called in the method `devType _open_o` (previously named `template_open_o`). The path have to be up to date

Create verification function

- For each htm file, create a .js with the same name.
It must be called in the script section of the htm on the top of the code. Use the same model as `data.js`
- For each input type parameter, create a test function with `unblur` call.

Define the parameters to be defined in ModelData

- Go to the _m file, in the function oDevice_WP_DFIG_internal_me () (previously named oDevice_template_internal_me), enter all the parameters that have to be defined in ModelData and assigned them a default value. The default value is given if the parameter is not defined.
- In the function save_data (previously template_save_o), give values to these parameters.
- In this function, keep:

```
FormData =this.my_device_services_.SaveParametersIntoFormData(this.Parameters)
```

```
var ModelData =
```

```
this.my_device_services_.SaveParametersIntoModelData(this.DerivedParameters,true);
```

These function populate the attributes FormData et ModelData

Documente the attributes availables in _i

- Go to the _i, and write down the attribute available in oDevice_devType_define_me to help other users to write scripts.