

Custom Output Data

If you want to add dynamic content to the clientarea template, the module provides an easy option without having to use WHMCS hooks or similar for integrating the content.

The function can be used to output data from the NOC-PS server properties or to display other content which you generate yourself via PHP.

How to use

- Navigate into the directory `/modules/servers/dedicated/`
- Open the file **customconfig.example.php** and copy the function "nocpsforwhmcs_customOutputData". Also copy the function_exists check.
- Open the file **customconfig.php**, or create the file if it does not exist and insert the copied code.
- Modify the PHP code accordingly, return an array and use the array in the module template file.

The module does pass 2 arguments to the nocpsforwhmcs_customOutputData function:

- \$params
- \$nocpsClient

You can access the returned array in the clientarea template file via the \$customOutputData variable. At the bottom of the page you find a code example.

Information about \$params

\$params contains the default WHMCS module parameters, which can be found [here](#). Additionally to this, the module extends the \$params array with following content:

Array Key	Information
\$params['nocps']['mac']	MAC address of the server
\$params['nocps']['permissions']	Information about the assigned ACL group
\$params['nocps']['moduleSettings']	Configuration of the module
\$params['nocps']['hostDetails']	Various information about the server

Use "logActivity (json_encode(\$params['nocps']));" to see the full content.

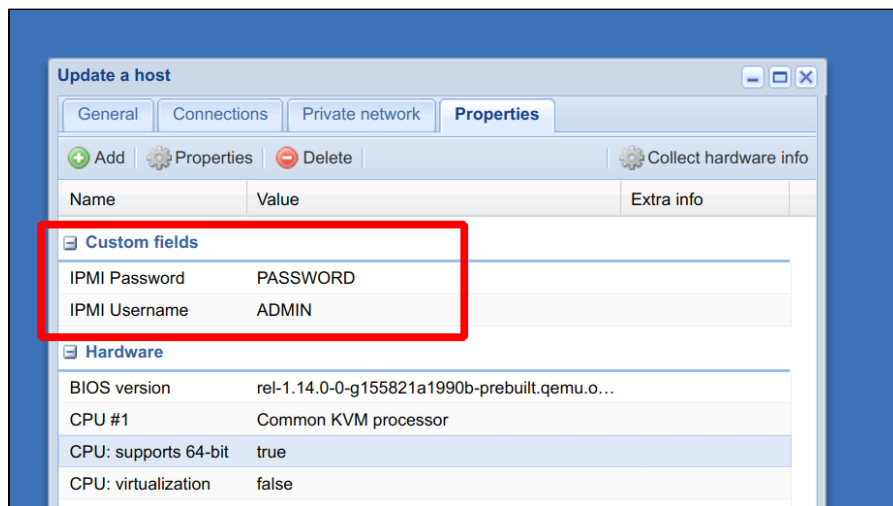
Information about \$nocpsClient

\$nocpsClient contains the NOC-PS API instance, which you can use to make API calls.

Usage example

In this example, the IPMI username and password is stored in the NOC-PS server properties. We get the details via the customOutputData function and integrate the values into the clientarea template.

NOC-PS Properties:



Content of customconfig.php:

```

<?php

if (!function_exists('nocpsforwhmcs_customOutputData')) {
    function nocpsforwhmcs_customOutputData ($params, $nocpsClient) {
        $returnFields = [ 'IPMI Username' => 'password', 'IPMI Password' => 'username' ];
        $return = [];

        try {
            $getHostProperties = $nocpsClient->getHostProperties ($params['nocps']['mac']);

            foreach ($getHostProperties['data'] as $res) {
                if (array_key_exists($res['name'], $returnFields)) {
                    $return[$returnFields[$res['name']]] = $res;
                }
            }

            // You must return an array:
            return [ 'ipmiLogin' => $return ];
        } catch (Exception $e) {
            logActivity ($e->getMessage());
        }
    }
}

```

Now the variable is usable in the smarty clientarea file of the module:

```

{if $customOutputData.ipmiLogin}
{* TO VIEW ALL VARS:
   {$customOutputData.ipmiLogin|@var_dump}
*}

<br />
IPMI Username: {$customOutputData.ipmiLogin.username.value}
<br />
IPMI Password: {$customOutputData.ipmiLogin.password.value}
{/if}

```

Instead of editing the original template of the module, you should make all changes via a template override file. More information about this feature is available here: [Template Override](#)