Console Extensions Development

https://pandorafms.com/manual/!775/
Permanent link:
https://pandorafms.com/manual/!775/en/documentation/pandorafms/technical_reference/07_anexo_console_extensions_development
2024/03/18 21:03

Console Extensions Development

We are working on the translation of the Pandora FMS documentation. Sorry for any inconvenience.

Console Extensions

Extensions are a form to develop new functionality for your Pandora Console as plugins.

In this article you will learn how to develop a extension.

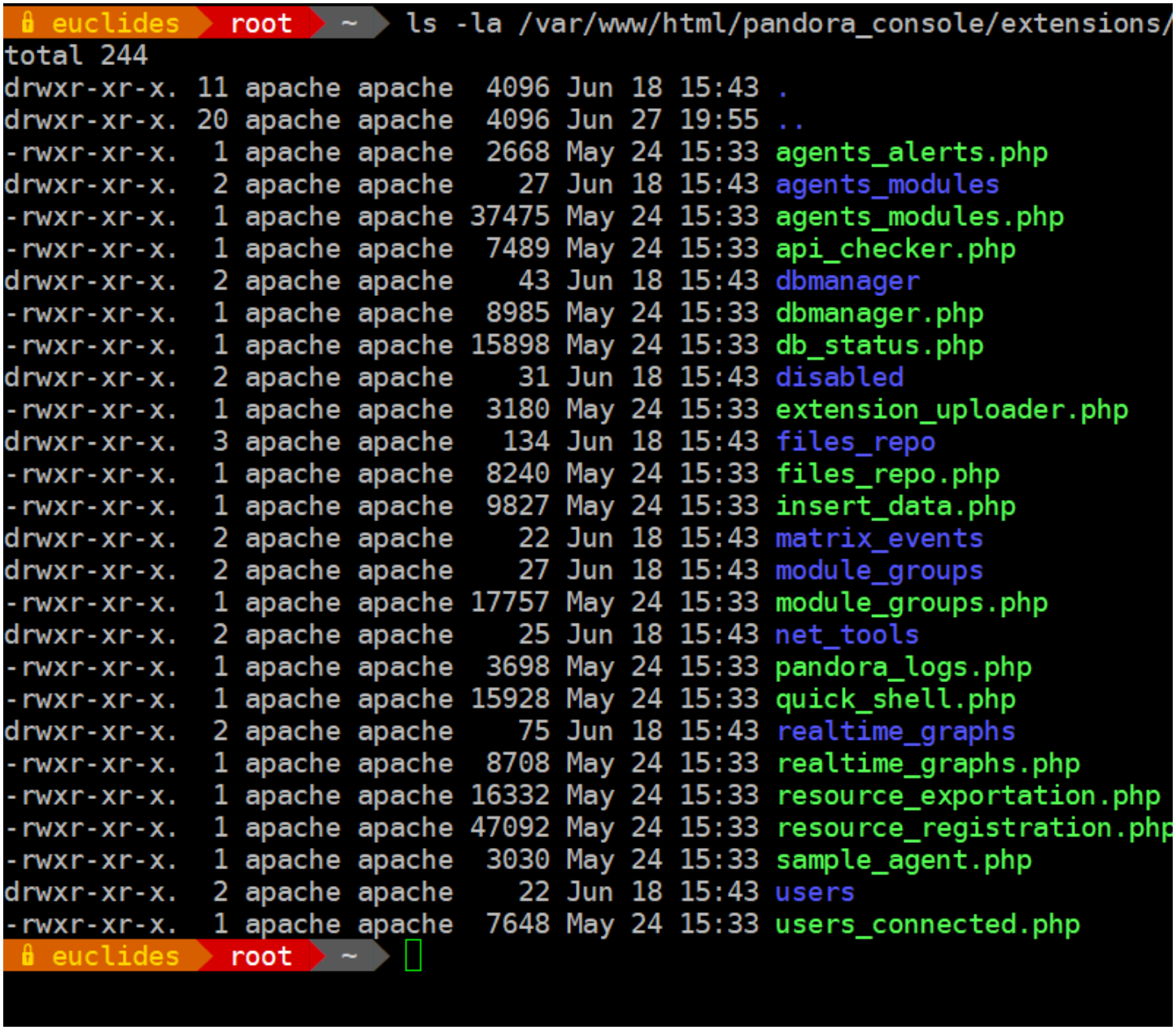
Kinds of Extensions

An extension can be one (or more) of the following types:

- Agent tab: Extensions that appear in the tab header of the agent's operation and/or edition view.
- Visible: Extensions that appear in the Pandora FMS Menu.
- Invisible: Extensions that are loaded and executed in the index.php of the Pandora FMS Menu but that do not appear in the PFMS Menu.

Directory of Extensions

The directory of extensions is a subdirectory in your local installation of Pandora FMS Console with the name `extensions`.



This directory contains for each extension the following:

Main file of the extension

This file has the code to load in Pandora FMS Console web.

Subdirectory of extension

this is optional and may contain the Icon image file (a 18 by 18 pixels image) to show next to the name of the extension in the menu and others files as translations, modules, images, etc.

Extension Skeleton

```
<?php

<Comments with license, author/s, etc...>
```

```
<php auxiliary code as functions, variables, classes that your extension use>

function <name of main function> () {
    <Main function Code>
}

/*-------------------------------------*/

/* Adds the link in the operation menu */
extensions_add_operation_menu_option ('<Name Extension>', '<father ID menu>',
'<relative path Icon>');

/* Adds the link in the godmode menu */
extensions_add_godmode_menu_option ('<Name Extension>', '<ACL level>', '<father
ID menu>', '<relative path Icon>')

/*-------------------------------------*/

/* Sets the callback function to be called when the extension is selected in the
operation menu */
extensions_add_main_function ('<name of main function>');

/* Sets the callback function to be called when the extension is selected in the
godmode menu */
extensions_add_godmode_function ('<name of godmode function>');
?>
```

API for Extensions

The API for extensions is stil under development and may change in the future.

The following sections contain the description of the functions in the API for extensions.

extensions_add_operation_menu_option

```
extensions_add_operation_menu_option ('<string name>', '<father ID menu>',
'<relative path Icon>')
```

This function adds the link to the extension with the given name in the Operations menu. The third parameter is optional and is the relative path to the icon image (18×18 pixels) to apear next to the link.

If this last parameter is not defined a plug icon ( ) will be used.

extensions_add_godmode_menu_option

```
extensions_add_godmode_menu_option ('<Name Extension>', '<ACL level>' , '<father ID menu>', '<relative path Icon>')
```

This function adds the link to the extension with the given name in the Godmode menu if the user has the required ACL level as indicated by the second parameter. The forth parameter is optional and is the relative path to the icon image (18×18 pixels) to apear next to the link.

If this parameter is not defined a plug icon ( ) will be used.

extensions_add_main_function

```
extensions_add_main_function ('<name of main function>')
```

Sets the callback function that will be called when the user clicks on the link to the extension in the operation menu

extensions_add_godmode_function

```
extensions_add_godmode_function ('<name of godmode function>')
```

Adds the extension function to call once the user goes to the extension in the Pandora FMS Web Console godmode instead of loading the main function.

extensions_add_login_function

```
extensions_add_login_function ('<name of login function>')
```

Adds the extension function to call once the user successfully logs into the Pandora FMS Web Console.

extensions_add_godmode_tab_agent

```
extensions_add_godmode_tab_agent('<ID of extension tab>', '<Name of extension tab>', '<Image file with relative dir>', '<Name of function to show content of godmode tab agent>')
```

Adds one more tab to the agent edit view that when it is selected executes the code of the name function that we pass to it.

extensions_add_opemode_tab_agent

```
extensions_add_opemode_tab_agent('<ID of extension tab>', '<Name of extension tab>', '<Image file with relative dir>', '<Name of function to show content of operation tab agent>')
```

Adds one more tab to the agent operating view than when it is selected will execute the code of the name function that we pass to it.

extensions_add_translation_string_function

```
extensions_add_translation_string_function('<Name of translation function>')
```

Sets the callback function that will be called from common translation function for extensions.

Father IDs in menu

List of available strings IDs for use in extension API. If use null value or not incluyed param in `call` function, the extension appear only in submenu of extension.

Operation

- `estado`: Monitoring view.
- `network`: Network view.
- `reporting`: Reporting and data visualization.
- `gismaps`: GIS view.
- `eventos`: Events view.
- `workspace`: User's workspace.

Administration

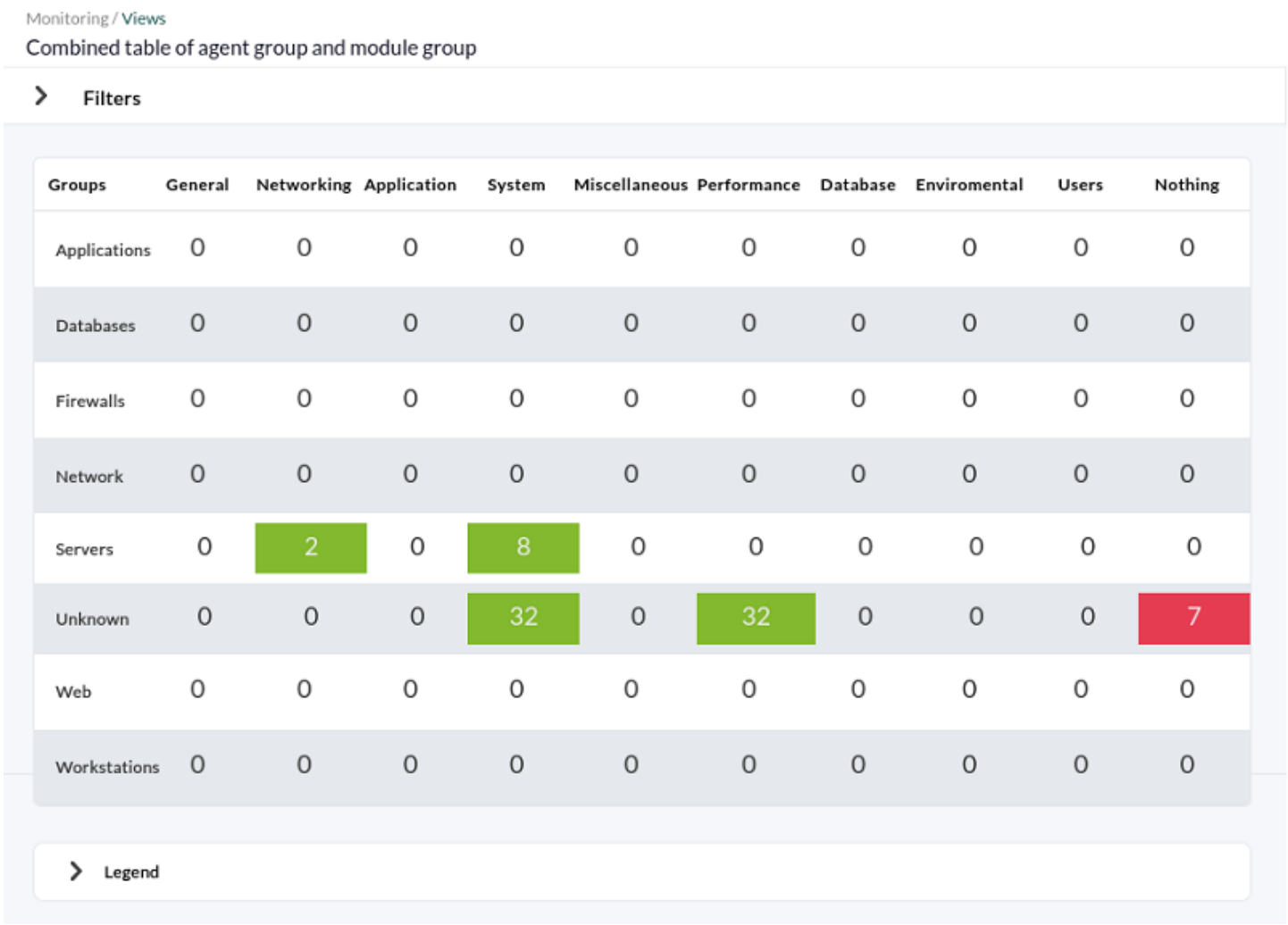
- `gagente`: Manage monitoring.
- `gmassive`: Massive operations.
- `gmodules`: Manage modules.
- `galertas`: Manage alerts.
- `gusuarios`: Manage users.
- `godgismaps`: Manage GIS.
- `gserver`: Manage servers.
- `glog`: System logs.
- `gsetup`: SetupConfiguración.
- `gdbman`: DB Maintenance.

Administration Enterprise

These elements are only available with Enterprise version.

- `gpolicies`: Manage policies.

Example



The extension show a table where the columns are Modules groups and the rows the Agent groups. And each cell have a colour with the next meanings:

- Green: When all modules of Group are OK.
- Yellow: When at least one monitor in warning.
- Red: At least one monitor fails.

And this extension hang from the Operation menu in Agents.

Source code

```
<?php
/**
 * Pandora FMS- http://pandorafms.com
```

```
 * ==================================================
 * Copyright (c) 2005-2009 Artica Soluciones Tecnologicas
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation for version 2.
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 */

/**
 * Extension specific translation function
 */
 function extensions_translation() {
   $translates = array(
     'es' => array(
       "sentence" => "translated sentence"
     ),
     /*
       ... for other language's definitions.
     */
   );

   $args = func_get_args();
   $string = array_shift($args);
   $user_language = get_user_language ();
   if(isset($translates[$user_language][$string])){
     return vsprintf($translates[$user_language][$string], $args);
   }
   else{
     return false;
   }
 }

/**
 * Translate the array texts using gettext
 */
 function translate(&$item, $key) {
     $item = ___($item);
 }

/**
 * The main function of module groups and the enter point to
 * execute the code.
 */
function mainModuleGroups() {
    global $config; //the useful global var of Pandora Console, it has many data
can you use

    //The big query
```

```
    $sql = "select COUNT(id_agente) AS count, estado
        FROM tagente_estado
        WHERE utimestamp != 0 AND id_agente IN
            (SELECT id_agente FROM tagente WHERE id_grupo = %d AND disabled IS
FALSE)
            AND id_agente_modulo IN
            (SELECT id_agente_modulo
                FROM tagente_modulo
                WHERE id_module_group = %d AND disabled IS FALSE AND
delete_pending IS FALSE)
        GROUP BY estado";

    echo "<h1>" . ___("Combine table of agent group and module group") .
"</h1>";

    echo "<p>" . ___("This table show in columns the modules group and for rows
agents group. The cell show all modules") . "</p>";

    $agentGroups = get_user_groups ($config['id_user']);
    $modelGroups = get_all_model_groups();
    array_walk($modelGroups, 'translate'); //Translate all head titles to
language is set

    $head = $modelGroups;
    array_unshift($head, ' ');

    //Metaobject use in print_table
    $table = null;
    $table->align[0] = 'right'; //Align to right the first column.
    $table->style[0] = 'color: #ffffff; background-color: #778866; font-weight:
bolder;';
    $table->head = $head;

    //The content of table
    $tableData = array();
    //Create rows and celds
    foreach ($agentGroups as $idAgentGroup => $name) {

        $row = array();

        array_push($row, $name);

        foreach ($modelGroups as $idModelGroup => $modelGroup) {
            $query = sprintf($sql,$idAgentGroup, $idModelGroup);
            $rowsDB = get_db_all_rows_sql ($query);

            $states = array();
            if ($rowsDB !== false) {
                foreach ($rowsDB as $rowDB) {
                    $states[$rowDB['estado']] = $rowDB['count'];
                }
```

```
            }

            $count = 0;
            foreach ($states as $idState => $state) {
                $count = $state;
            }

            $color = 'transparent'; //Defaut color for cell
            if ($count == 0) {
                $color = '#babdb6'; //Grey when the cell for this model group
and agent group hasn't modules.
                $alinkStart = '';
                $alinkEnd = '';
            }
            else {
                $alinkStart = '<a
href="index.php?sec=estado&sec2=operation/agentes/status_monitor&status=-1&ag_gr
oup=' . $idAgentGroup .
                    '&modulegroup=' . $idModelGroup . '">';
                $alinkEnd = '</a>';

                if (array_key_exists(0,$states) && (count($states) == 1))
                    $color = '#8ae234'; //Green when the cell for this model
group and agent has OK state all modules.
                else {
                    if (array_key_exists(1,$states))
                        $color = '#cc0000'; //Red when the cell for this model
group and agent has at least one module in critical state and the rest in any
state.
                    else
                        $color = '#fce94f'; //Yellow when the cell for this
model group and agent has at least one in warning state and the rest in green
state.
                }
            }

            array_push($row,
                '<div
                    style="background: ' . $color . ' ;
                        height: 15px;
                        margin-left: auto; margin-right: auto;
                        text-align: center; padding-top: 5px;">
                    ' . $alinkStart . $count . ' modules' . $alinkEnd .
'</div>');
        }
        array_push($tableData,$row);
    }
    $table->data = $tableData;

    print_table($table);

    echo "<p>" . ___("The colours meaning:") .
```

```
        "<ul>" .
        '<li style="clear: both;">
            <div style="float: left; background: #babdb6; height: 20px; width:
80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Grey when the cell for this model group and agent group hasn't
modules.") . "</li>" .
        '<li style="clear: both;">
            <div style="float: left; background: #8ae234; height: 20px; width:
80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Green when the cell for this model group and agent has OK state
all modules.") . "</li>" .
        '<li style="clear: both;"><div style="float: left; background: #cc0000;
height: 20px; width: 80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Red when the cell for this model group and agent has at least
one module in critical state and the rest in any state.") . "</li>" .
        '<li style="clear: both;"><div style="float: left; background: #fce94f;
height: 20px; width: 80px;margin-right: 5px; margin-bottom: 5px;"> </div>' .
            ___("Yellow when the cell for this model group and agent has at
least one in warning state and the rest in green state.") . "</li>" .
        "</ul>" .
        "</p>";
}

extensions_add_operation_menu_option("Modules groups", 'estado',
'module_groups/icon_menu.png');
extensions_add_main_function('mainModuleGroups');
extensions_add_translation_string_function('extensions_translation');
?>
```

Explanation

In the source code there are two parts:

- The source code of extension.
- The API calls functions.

The order of parts is indifferent, but it's better put “The API calls functions” in the bottom of you main file extension because the style guidelines advise to add this part into bottom thus all extensions have more or less a same style.

Source code of extension

In this case for this example has three function in the same file, but if you has a complex code, it is better divide in many files (and it save in subdirectory extension), the functions are:

- Function `extensions_translation()`
  This function is for extension specific translation. It is called from the common translation function.
- Function `translate(&$item, $key)`

This function use for callback in the array_walk function. Because the main function keep the titles of columns and the titles of rows in array without translations.

- Function `mainModuleGroups()`
  This is the heard of extension, and it's huge of lines, I see not all code, see some important parts:
  - The first is access to config global var. In this var has many configurations and default values for many things of Pandora FMS Console.
  - The second var is the query in MySQL into a string. And the `%d` is the format placeholders is for Id Group and Id Module Group and these are sustitute for value in sprintf function into foreach loops.
  - Some `echos` for print the text before the table.
  - Extract of DB two arrays with one dimension and the index is id, and the content is title for columns (Module groups) and rows (Agent group) in each case.
  - Translate the Model Group array titles.
  - Make the meta-object $table, fill by rows and print.
    - Before the `foreach ~loops`, define into `$table` the head and styles of table.
    - The first loop is for rows (each agent group).
      - The second loop is for columns in current row (each model group).
        - Then for each cell, it has two number, id model group and id agent group, with this two number we make a query to database and we obtain the files.
        - Proccess the result array for obtain other array that is array and the index is a integer of diferents kinds of monitor states and the content is a count of monitor in this state.
        - Well, the only thing left is to make or fill the content of cell in HTML. The trick is easy. If the count off all states is zero, the background for div in CSS is grey. If `$states[1] != 0` or in human language there is one at least of monitor in critical state, the `div` has a red color. If the array only have a one cell and it's the normal state, the green is in this `div`. And others cases, the yellow is the color for `div`.
        - Add link in the cell if count is more than `0`.
        - Save the row in `$table`, and start other iteration of foreach.
  - Print the table.
  - Print the legend and other notes in the bottom of page.

API calls functions

There are only a few lines of code. The operations in these lines are:

- Insert the extension into Pandora FMS menu.
- And it's with the call

```
extensions_add_operation_menu_option("Modules groups", 'estado',
'module_groups/icon_menu.png');
```

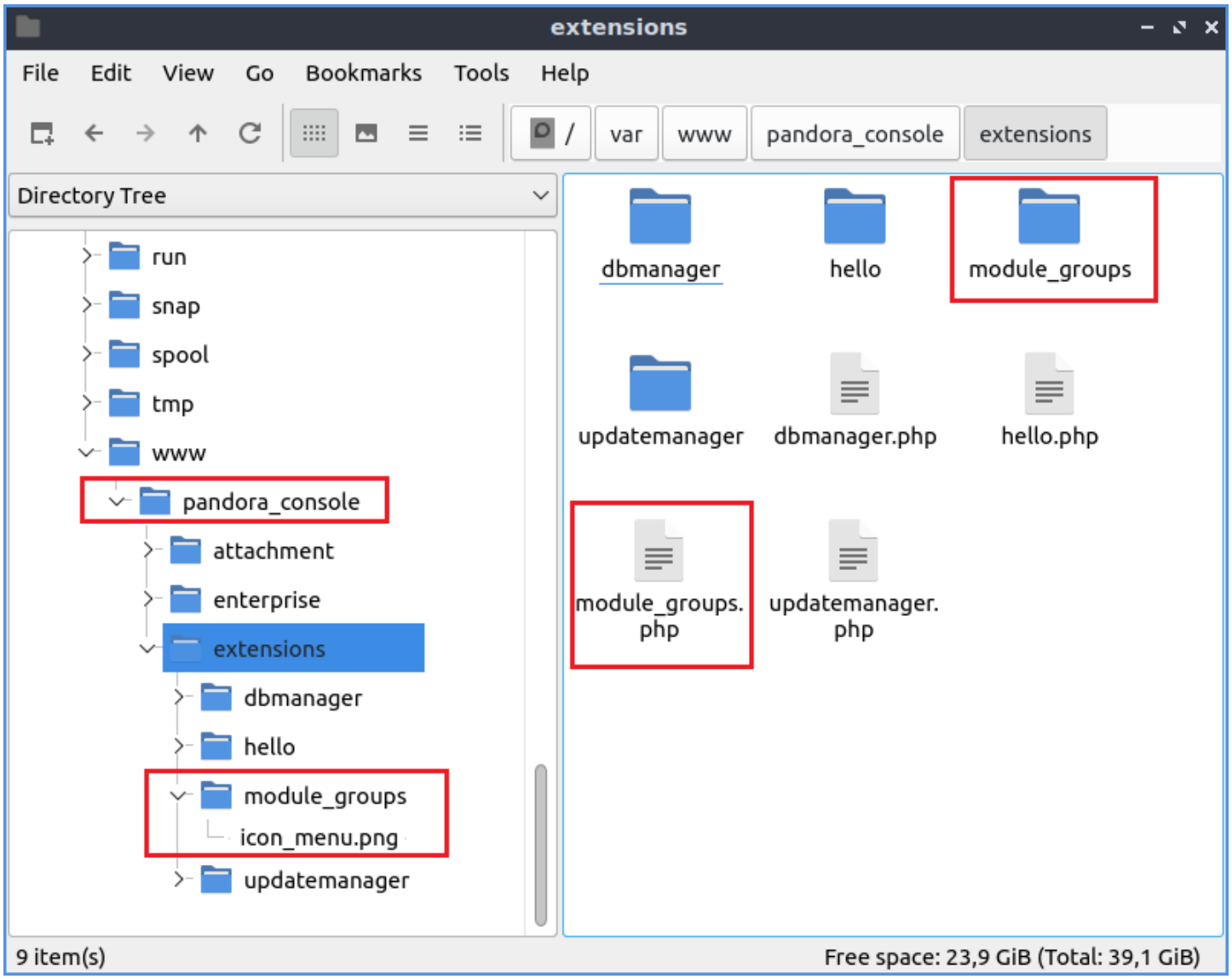
Where:

- 'Modules groups' is the name appear in submenu of agents.
- 'Status' is the element that hangs from the extension.
- 'module_groups/icon_menu.png' is the image icon appear in submenu, the path is relative to your extension directory.
- Define the main function of this extension .

And it's with the call `extensions_add_main_function('mainModuleGroups');` where:
- 'mainModuleGroups' is the name of extension main function.

The order in which the functions are named is irrelevant.

Directory organization



The instalation of extension is very easy, because the Pandora FMS Console search new extensions and add into system when new extension is found. You only copy all files of extensions into the directory extension in your Pandora FMS Console instalation. But you must set the permissions for the Pandora FMS Console can read the files and subdirectories of extension.

In the screenshot, the extension has a directory structure:

- `module_groups`
  - `icon_menu.png`
- `module_groups.php`

And the extension directory is for example in `/var/www/pandora_console`.

Subdirectory

In this case, the example has one subdirectory, and usually any extension must has one subdirectory. The subdirectory has the same name as the name extension and the main file. The subdirectory of the example only has an image icon file (`icon_menu.png`). This icon is shown in the Pandora FMS Menu.

Go back to Pandora FMS documentation index