HOW TO CUSTOMIZE NETSHe?

Programming languages

NETSHe was written in php (compatible with version 5), xhtml, shell (*sh*) and JavaScript.
It uses some third party software as described in the User manual.

Application Layout

NETSHe keeps all own code under */opt/rb* directory. It places shell- and binary-executables in */opt/rb/bin* directory, php-executables in */opt/rb/php* directory. Some configuration files (like *php.ini*), startup-scripts and functions for shell files are located in */opt/rb/etc* directory.
Main configuration file is placed in */etc/.ssxapp* directory and is named *.main.conf* . This directory will be created automatically on first system boot, if it does not exist.
Configuration file with default (factory) settings is placed in */opt/rb/conf.default/.ssxapp* directory. It is named *.main.conf* and is copied to */etc/.ssxapp* directory on first system boot or when 'factory settings' are chosen in the web-interface.
Configuration file has an XML-format.

Php-libraries are placed in */opt/rb/lib/php* directory and portions of them are placed in other directories depending on the operating system and the distributive (e.g. */opt/rb/lib/php/linux*).
Web-interface files (html-templates, JavaScript-files, etc) are placed into */opt/rb/www* directory.
NETSHe has a MVC-like (Model-View-Controller) architecture. NETSHe has the core system and an extensible additional system.
'Model' partition performs data transformation in accordance with the user actions. For example, it queries data from databases and/or other storages, transfers data to 'Controller' level, receives data back and stores modified data into storages. Model files are placed into */opt/rb/lib/php/model* directory.
'View' partition helps to display data and actions with regards to system data, user actions and 'Controller' actions. This partition includes html-templates to display information and internationalization abilities. Html-templates are placed into */opt/rb/www/view* directory. Internationalization files are placed into */opt/rb/lib/php/i18* directory.
'Controller' partition performs the main action: it reacts to user actions, forms output (through 'View' level) and modifies data (through 'Model' level). Controller files are placed into */opt/rb/lib/php/controller* directory.
The core provides basic system functionality (authorization, central configuration storage, etc) and functionality of MVC partitions (e.g. it finds and loads the required internationalization and model files automatically).

A basic term in NETSHe is an 'application'. Every 'application' has its own name and a group of files (executables, controllers, models, views and etc). An 'application' has at least one own file. All files that form an 'application' are placed into directories with the name of this 'application'. For example, an application 'CP' (Control Panel) has many files, which are grouped into directories */opt/rb/lib/php/controller/cp*, */opt/rb/lib/php/model/cp*, */opt/rb/lib/php/i18/cp*, */opt/rb/www/views/cp*, */opt/rb/www/app/cp*.
Typically, an application performs at least one action or one job (e.g. manages *iperf* daemon, controls settings, starts and stops daemon). We divide all code and other files related to this job into some portions:

- php-code, that represents the main logic of an application (controller) is placed into file */opt/rb/lib/php/controller/cp/class.iperf.inc.php*
- html-template is placed into */opt/rb/www/views/cp/iperf.tpl.html*

- translation file is placed into */opt/rb/lib/php/i18/cp/en/iperf.str*. Where *en* means English language.

Have a look at the corresponding files:

/opt/rb/lib/php/controller/cp/class.iperf.inc.php

```
<?php

require_once(dirname(__FILE__).'/../../util-lib.inc');

class iperf
{
        var $owner = '';
        var $classname = 'iperf';
        var $can_access = false;
        var $can_add = false;
        var $can_edit = false;
        var $can_remove = false;

        function iperf( $app) {
                $this->owner = $app;
                $this->owner->set_help( $this->classname);
                $this->can_access = ValidateLevel( $this->owner->config['SECURITY']['SYSTEM_OPERATORS']);
                $this->can_add = ValidateLevel( $this->owner->config['SECURITY']['SYSTEM_OPERATORS']
+LVL_ADD);
                $this->can_edit = ValidateLevel( $this->owner->config['SECURITY']['SYSTEM_OPERATORS']
+LVL_EDIT);
                $this->can_remove = ValidateLevel( $this->owner->config['SECURITY']['SYSTEM_OPERATORS']
+LVL_REMOVE);
        }

        function config() {
            if ( !$this->can_access) {
                $this->owner->permission_denied();
                return;
            }
            $t = $this->owner->load_template_resource(array('hForm'=>'iperf'));

            $t->set_var('VAL_FORMACTION', Linkage('', 'q=cp.iperf.save'));
            // Labels
            $t->set_var('LS_PATH', $this->owner->get_ls_path());
            $t->set_var('BTN_SAVE', STR_CMMN_SAVE);
            $t->set_var('TXT_APPLY', STR_CMMN_APPLY);
            $t->set_var('TXT_HIGHLIGHTEDNOTE', STR_CMMN_HIGHLIGHTEDNOTE);
            $this->owner->page_title = STR_IPERF_TITLE;
            $tbLink = '';
            if ( $this->can_edit) {
                if ( iperf_status())
                  $tbLink .= '<a href="' . Linkage('', 'q=cp.iperf.iperf_stop') . '">' . GetImage('process-stop.png') .
'&nbsp;' . STR_IPERF_STOP . '</a><a href="' . Linkage('', 'q=cp.iperf.iperf_restart') . '">' . GetImage('reload.png') .
'&nbsp;' . STR_IPERF_RESTART. '</a>';
                else
                  $tbLink .= '<a href="' . Linkage('', 'q=cp.iperf.iperf_start') . '">' . GetImage('start-here.png') .
'&nbsp;' . STR_IPERF_START. '</a>';
            }
            $this->owner->print_bbar( $tbLink);
            $t->set_var('TXT_ENABLE', STR_IPERF_ENABLE);
            $t->set_var('TXT_DESC', STR_IPERF_DESC);
            if ( empty( $this->owner->config['services']['iperf']))
                $t->set_var('VAL_CHECKED', '');
            else
                $t->set_var('VAL_CHECKED', 'checked="checked"');
            $t->pparse('out', 'hForm');
        }

        function save() {
            if ( !$this->can_edit) {
                $this->owner->permission_denied();
                return;
            }
            if ( isset( $GLOBALS['item_enable']))
                $this->owner->config['services']['iperf'] = 1;
            else
                $this->owner->config['services']['iperf'] = 0;
            if ( $this->owner->save()) {
                if ( isset( $GLOBALS['apply']))
                    iperf_restart();
                Refresh('q=cp.iperf.config');
            } else $this->owner->error_msg(STR_CMMN_ERROR);
        }

    function iperf_stop() {
        if ( !$this->can_edit) {
            $this->owner->permission_denied();
            return;
        }
        iperf_stop();
        Refresh('q=cp.iperf.config');
    }

    function iperf_start() {
        if ( !$this->can_edit) {
            $this->owner->permission_denied();
            return;
        }
        iperf_start();
        Refresh('q=cp.iperf.config');
    }

    function iperf_restart() {
        if ( !$this->can_edit) {
            $this->owner->permission_denied();
            return;
        }
        iperf_restart();
        Refresh('q=cp.iperf.config');
    }

}
?>
```

/opt/rb/www/views/cp/iperf.tpl.html

```
<script type="text/javascript" language="JavaScript1.2" src="{LS_PATH}js/formcheck.js"></script>

<script type="text/javascript" language="JavaScript">
//<![CDATA[
function validateAndSubmitForm(form) {
    form.submit();
}

//]]>
</script>
<form method="post" action="{VAL_FORMACTION}" name="calform" id="calform">
```

```
    <table>
            <tr>
         <td colspan="2" align="left"><span class="comment">{TXT_DESC}</span><hr /></td>
        </tr>
        <tr>
          <td colspan="2">&nbsp;</td>
        </tr>
              <tr>
               <th>{TXT_ENABLE}&nbsp;:</th>
               <td>&nbsp;<input type="checkbox" id="item_enable" name="item_enable" value="0"
{VAL_CHECKED} /></td>
               </tr>
        <tr>
          <td colspan="2">&nbsp;</td>
        </tr>
        <tr>

          <td align="center">{TXT_APPLY}&nbsp;:&nbsp;<input type="checkbox" name="apply" value="0"
/></td><td align="center"><input onclick="javascript: return validateAndSubmitForm(this.form);"
value="{BTN_SAVE}" type="button" /></td>

        </tr>
        <tr>

          <td colspan="2">&nbsp;</td>

        </tr>
        <tr>

          <td colspan="2" align="center"><span class="highlight">{TXT_HIGHLIGHTEDNOTE}</span></td>

        </tr>

    </table>
  </form>
```

/opt/rb/lib/php/i18/cp/en/iperf.str

```
<?php
define("STR_IPERF_ENABLE", "Enable IPerf server (network perfomance measurement tool) ?");
define("STR_IPERF_DESC", "You can use built in tool to measure network perfomance");
define("STR_IPERF_START", "Start IPerf server");
define("STR_IPERF_RESTART", "Restart IPerf server");
define("STR_IPERF_STOP", "Stop IPerf server");
?>
```

It's a good idea to place shared files (library, templates, etc) into the directories '*core*'.

Have a look at the application 'Control Panel' (cp).

The web-interface begins as follows (/opt/rb/www/app/cp/index.php):

```
<?php

error_reporting(E_ALL);

require_once(dirname(__FILE__).'/../../../lib/php/controller/core/class.Application.inc.php');

class Root extends Application
{
            function Root()    {
                    $this->Application('SysUser', 'cp');
                    $this->initial_action = 'core.EntryUs.login';
                    if ( !isset( $this->config['services']) || !isset( $this->config['services']['first']) || empty( $this-
>config['services']['first']))
                       $this->default_action = 'cp.wizard';
                    else
                       $this->default_action = 'cp.begin';
          }
}

$app = new Root;
$app->exec();

?>
```

It is not a difficult code, is it?
We will return to the code later and describe the translation process first.

How to translate NETSHe into other languages?

NETSHe supports multiple languages. Basically, it has at least one translation (default language as defined in the main configuration file) and none or more additional translations (should be defined in the main config too).
So, the default language and available translations are defined in *.main.conf* as follows:

```
<LANGUAGES>en,ru</LANGUAGES>
<en>English</en>
<ru>Русский</ru>
<LANGUAGE>en</LANGUAGE>
```

We have two languages enabled (English and Russian) and we have English as the default language.
Note: NETSHe will show all messages displayed in the console in the default language. Also, when you've connected to the web-interface, the latter checks the preferred language, which is enabled in your browser. NETSHe displays the start page in this preferred language, if the relevant translation is available. Otherwise, it displays the start page in the default language.

How to make a translation?

Requirements:

1. English translation (or other available and understandable language).
2. Text editor that supports UTF-8 encoding (without BOM).

Note: your translated file must be UTF-8 encoded without BOM

For example, to translate NETSHe into Spanish, you must perform the following steps:

1. Create */opt/rb/lib/php/i18/core/es* directory.
2. Open the file */opt/rb/lib/i18/core/en/ssxapp.str* in an editor, which meets the above requirements.

```
<?php
         define('STR_APP_FORGOT', 'Forgot password');
         define('STR_APP_FORGOT_SUBJ', 'Letter from system administrator');
         define('STR_APP_FORGOT_EMAIL', 'Dear %s.\n\n - %s\n\nRegards, administrator');
         define('STR_APP_ACCESSDENIED', 'Access denied');
         define('STR_APP_VALID_LOGIN', 'Access permited.');
         define('STR_APP_INVALID_PASSWORD', 'Invalid password');
         define('STR_APP_INVALID_LOGIN', 'Invalid username');
         define('STR_APP_LOGIN', 'Entrance');
         define('STR_APP_HELP', 'Help');
```

```
	define('STR_APP_LOGOUT', 'Logout');
	define('STR_APP_CLEAR', 'Clear');
	define('STR_APP_LOGIN_WELCOME', 'Authorization');
	define('STR_APP_LOGIN_RULES', 'Attention! All your action stores! Enter Login and Password for
authorization.');
	define('STR_APP_LOGIN_ACTION', 'Log in');
	define('STR_APP_USERNAME', 'Username');
	define('STR_APP_PASSWORD', 'Password');
	define('STR_APP_LANGUAGE', 'Interface language');
	define('STR_APP_DOMAIN', 'Domain');
	define('STR_APP_PROVIDER', 'Provider');
	define('STR_APP_DBERR', 'Error in database-susystem was encountered!');
	define('STR_APP_HOME', 'Home');

?>
```

3. Edit file as follows.

```
<?php
	define('STR_APP_FORGOT', 'Recordar contraseГ±a');
	define('STR_APP_FORGOT_SUBJ', 'Mensaje del administrador del sistema');
	define('STR_APP_FORGOT_EMAIL', 'Estimado %s.\n\n - %s\n\nSaludos cordiales, administrator');
	define('STR_APP_ACCESSDENIED', 'Acceso denegado');
	define('STR_APP_VALID_LOGIN', 'Acceso permitido.');
	define('STR_APP_INVALID_PASSWORD', 'ContraseГ±a invГЎlida');
	define('STR_APP_INVALID_LOGIN', 'Nombre de usuario invГЎlido');
	define('STR_APP_LOGIN', 'Ingreso');
	define('STR_APP_HELP', 'Ayuda');
	define('STR_APP_LOGOUT', 'Salida');
	define('STR_APP_CLEAR', 'Limpiar');
	define('STR_APP_LOGIN_WELCOME', 'AutorizaciГіn');
	define('STR_APP_LOGIN_RULES', 'AtenciГіn! Todas sus acciones serГЎn guardadas! Ingrese su nombre de
usuario y contraseГ±a para ser autorizado.');
	define('STR_APP_LOGIN_ACTION', 'Ingresar');
	define('STR_APP_USERNAME', 'Nombre de usuario');
	define('STR_APP_PASSWORD', 'ContraseГ±a');
	define('STR_APP_LANGUAGE', 'Lenguaje de la interfaz');
	define('STR_APP_DOMAIN', 'Dominio');
	define('STR_APP_PROVIDER', 'Proveedor');
	define('STR_APP_DBERR', 'Se encontrГі un error en la base de datos-susystem!');
	define('STR_APP_HOME', 'Principal');
?>
```

4. Save file as */opt/rb/lib/php/i18/core/es/ssxapp.str*
5. Translate all related files as described above.
6. Edit *.main.conf* as follows.

```
<LANGUAGES>en,es,ru</LANGUAGES>
<en>English</en>
<es>Spanish</es>
<ru>Русский</ru>
<LANGUAGE>en</LANGUAGE>
```

Enjoy the new language!