

# PLESK7 RELOADED

## CREATING AND INSTALLING CUSTOM PLESK SKINS

---

Distribution of this work or derivative of this work in any form is prohibited unless prior written permission is obtained from the copyright holder.

Linux is a registered trademark of Linus Torvalds. RedHat is a registered trademark of Red Hat Software, Inc. All other trademarks and copyrights are the property of their respective owners.

13800 Coppermine Road, Suite 112, Herndon, VA, 20171 USA, Ph.: 703 815-5670, Fax.: 703 815-5675

---

---

## Table of Contents

1. Introduction .....	1
What Is a Skin? .....	1
Areas of the User Interface .....	1
Files That Compose a Skin .....	2
Structure Of Skin Directory .....	4
2. Creating a Skin .....	5
Creating the Skin Directory .....	5
Customizable Properties .....	6
Preparing a Skin Package for Uploading to the Control Panel .....	22
3. Installing a Skin .....	25
Installing the Skin RPM Package Using the Command Line .....	25



---

# Chapter 1. Introduction

This document is a guide to creating and installing *skins* - custom interface appearance styles - for Plesk. Here you can find the structure of skin directories as well as its contents description, instructions on how to create your own custom skin and how to install it and make useable on your server.

## What Is a Skin?

In Plesk a skin is in fact a set of CSS and image files. The CSS files define the style of the Plesk interface elements; the image files are the Plesk interface icons, logotype images and other pictures, used in CSS files. All these files, placed in corresponding sub-directories, compose the structure of the *skin directory*.

### **IMPORTANT**

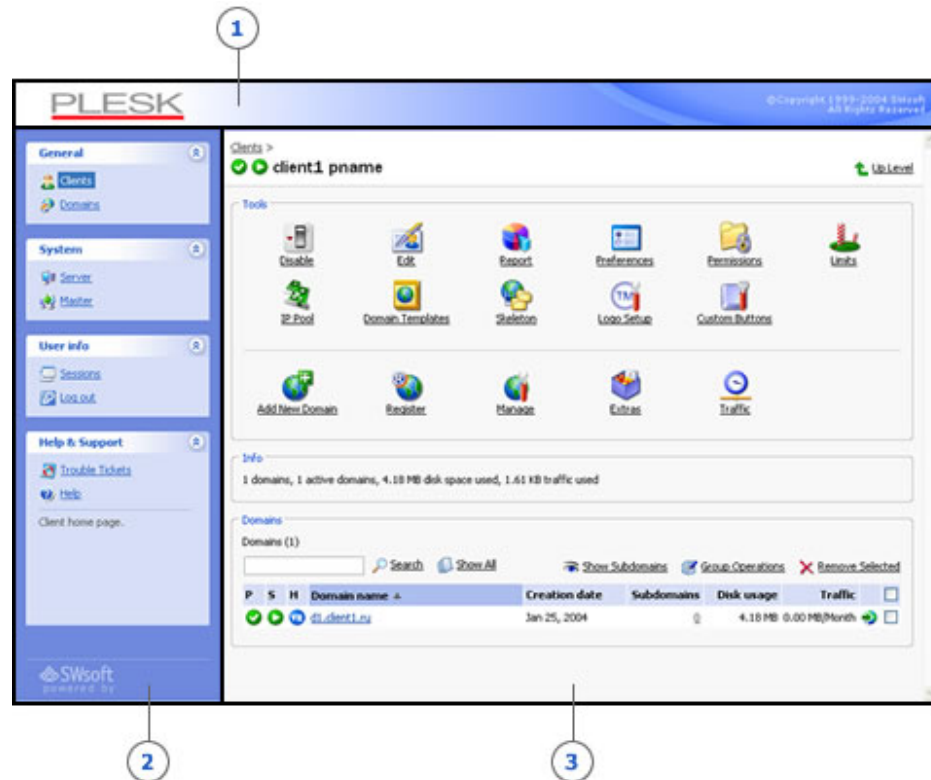
The development and/or modification of a skin require a strong knowledge of Cascading Style Sheets [<http://www.w3.org/Style/CSS/>]. It is absolutely necessary due to the fact that the process of creating a custom skin is largely a matter of editing selectors in CSS files.

Skins are an easy and flexible way to diversify your Plesk user interface appearance. Using skins you can change the colors of the interface areas, set new fonts properties, use different images for icons in the interface, etc.

It takes only a few clicks to replace one skin with another. Different skins can be used by different users at one server.

## Areas of the User Interface

The Plesk user interface can logically be split into three parts: *top area*, *left (navigation) area* and *main area*.



1. *top area* contains the logotype image
2. *left (navigation) area* contains navigation items and context help area
3. *main area* contains the groups of available operations (based on current context), input forms, lists, and other similar interface elements

Each such area allows for individual customization of appearance within a skin.

## Files That Compose a Skin

### custom.css and layout.css

Each interface area has the corresponding two CSS files describing its appearance:

- *custom.css* contains selectors for visual properties (color, font, etc.) of the user interface elements
- *layout.css* contains selectors that define the layout of the user interface elements

#### **i** NOTE

The option of modifying *layout.css* is recommended only for the advanced CSS designers.

The *custom.css* and *layout.css* in the *help/* directory within the skin directory

define the appearance of the Help pages.

## **buttons.css**

Additionally the main area uses file *buttons.css*, which defines the appearance of certain buttons in the user interface. For example, in the *XP-skins* it defines what images are used for the icons in the Tools groups.

This file is not a requirement and is not needed if the appearance of multiple buttons is not redefined in the skin. *buttons.css* is addressed from *main/custom.css*, its contents were separated only for the sake of ease of use.

## **general.css**

The file *general.css* contains style settings general for all interface elements. The style specified here will be applied when displaying an interface element unless it was specifically redefined for the corresponding area of the user interface.

## **tabs.css**

The file *tabs.css* contains style settings, which define appearance of tab elements.

## **info.xml**

When you access the Skin Properties page within the Skins repository in Plesk, it displays the information on skin, such as author's name, creation date, screenshots and descriptions. This information is stored in the *info.xml* file, which is located in the skin directory. The thumbnails and screenshots are stored in the *screenshots/* directory.

## **Image files**

Image files are stored in three directories:

- *icons/* contains image files required for the user interface (state/status icons, list operations, etc.)
- *images/* contains image files used with the specific skin for customizing elements, set of these can be different for different skins. Links to these images are provided in the CSS files
- *screenshots/* contains control panel screenshots and thumbnails.

## Structure Of Skin Directory

The skin directories are located in `/usr/local/psa/admin/htdocs/skins/`

The structure of the skin directory:

- `custom_skin/` - a custom skin directory
  - `css/` - all CSS files
    - `top/`
      - `custom.css`
      - `layout.css`
    - `left/`
      - `custom.css`
      - `layout.css`
    - `main/`
      - `custom.css`
      - `layout.css`
      - `tabs.css`
    - `help/`
      - `custom.css`
      - `layout.css`
    - `general.css`
  - `icons/` - all of the Plesk interface icons
  - `images/` - all image files, referenced in the CSS files
  - `screenshots/` - screenshots and thumbnails, referenced in the skin description file
  - `info.xml` - the file that stores the skin description

---

## Chapter 2. Creating a Skin

This chapter provides instructions on how to compose a new skin and prepare a skin package for uploading to the control panel. The first main step in this process is the creation of the structure of the skin directory along with all its files, another is the actual customization of the visual appearance of elements of the user interface by editing the properties in the corresponding CSS files of the skin. The last step is preparing a skin package.

### Creating the Skin Directory

In order to create the skin directory you need to create the structure of directories along with the corresponding CSS files as described in the section Structure Of Skin Directory. When this task is complete, you can proceed to editing the style properties.

### Using Existing Skin As Template

In order to speed up and simplify the creation of the skin directory you can make use of an already existing skin (one of the default ones) installed on your server as a template for your own custom skin.

Create your future skin directory (e.g. `my_skin`):

```
# mkdir ~/my_skin
```

Copy to this directory one of the default skins:

#### NOTE

The directory where all skins are located in Plesk is  
`/usr/local/psa/admin/htdocs/skins/`.

```
# cp -r /usr/local/psa/admin/htdocs/skins/winxp.blue/*  
~/my_skin
```

At this point you will have in your skin directory (`~/my_skin/`) the complete skin directory structure along with the CSS and image files of the Plesk default skin `winxp.blue`.

The skin template is ready to be used. Now you can begin editing the CSS files and adding in the necessary image files within the skin directories creating your unique style of Plesk user interface appearance.

## Customizable Properties

Every type of element of the user interface that can be customized is described by the corresponding selectors within the appropriate CSS files of the skin. This section considers the areas of the user interface and gives the listing of these selectors and elements they correspond to, as well as provides simple examples of using these selectors in the CSS files.

### General

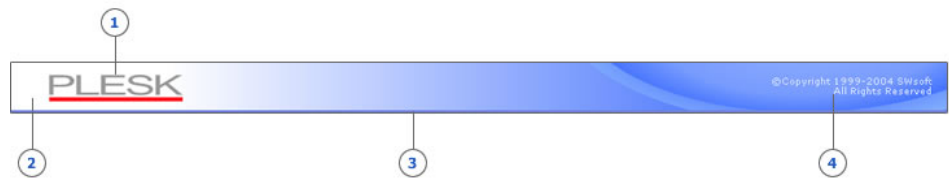
The properties that are common for all areas of interface, assigned in the file *general.css*.

**Table 2.1. General properties**

UI Element	Selector	CSS code sample
common background, font	body, td, th	<pre>body {   font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;   font-size: 11px;   font-weight: normal;   color: #000000;   background-color: #f6f6f6; }  td, th {   font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;   font-size: 11px; }</pre>
form elements	input, select, textarea	<pre>input, select, textarea {   font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;   font-size: 11px; }</pre>
link	a	<pre>a:link, a:visited, a:hover {   color: #0240a3; }</pre>

Further, for the specific elements, these properties can be redefined as desired.

## Top Area



1. *logotype image*
2. *background*
3. *frame separator line*
4. *top left image*

**Table 2.2. Top area properties**

N	Selector	CSS code sample
1.	can be set through the user's interface, but the default image is contained in the skin ( <i>images/def_plesk_logo.gif</i> )	-
2, 3.	body	<pre>body {   background-color: #ffffff;   background-image:   url(../images/top_bg.jpg);   background-repeat: repeat-x;   background-position: left bottom; }</pre>
4.	.body	<pre>.body {   background-image:   url(../images/top_body_bg.jpg);   background-repeat: no-repeat;   background-position: top right; }</pre>

## Left Navigation Area



1. *background*
2. *navigation sections header background*
3. *navigation sections header*
4. *expand/collapse navigation section*
5. *navigation section area*
6. *navigation item*
7. *selected navigation item*
8. *logged in user info*
9. *context help*
10. *'powered by' logotype image*

**Table 2.3. Left navigation area properties**

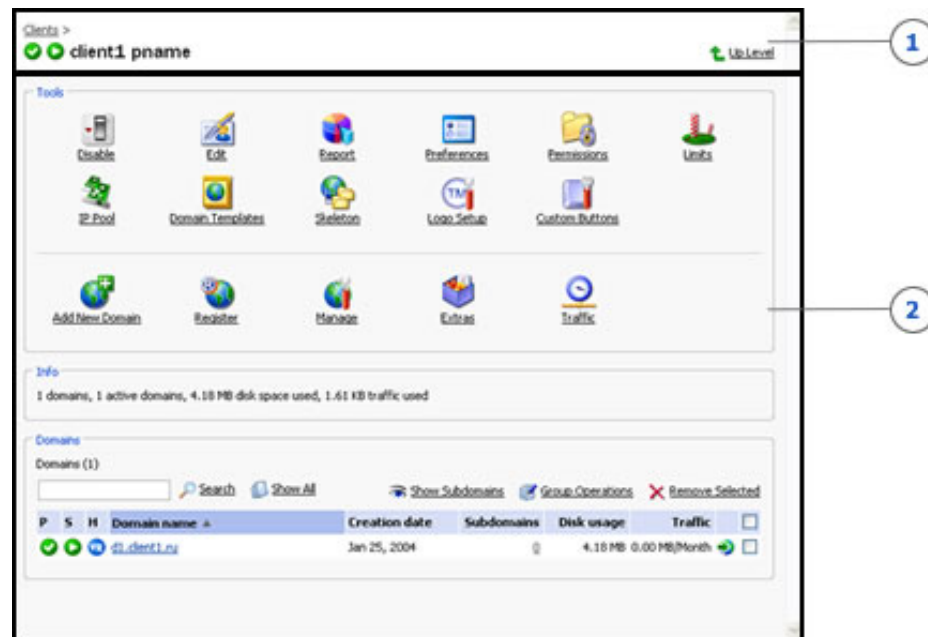
N	Selector	CSS code sample
1.	body	<pre>body {   background-color: #6e89dd; }</pre>
2.	.navOpened .navClosed	<pre>.navOpened, .navClosed {   background-color: #ffffff; }</pre>

N	Selector	CSS code sample
3.	<pre>.navOpened .navTitle .titleText .navClosed .navTitle .titleText</pre>	<pre>.navOpened .navTitle .titleText, .navClosed .navTitle .titleText {   color: #215dc6; }</pre>
3.	<p>Mouse over</p> <pre>.navOpened .navTitleOver .titleText .navClosed .navTitleOver .titleText</pre>	<pre>.navOpened .navTitleOver .titleText, .navClosed .navTitleOver .titleText {   color: #428eff; }</pre>
4.	<pre>.navTitle .titleHandle</pre>	<pre>.navTitle .titleHandle {   background-color: #215dc6; }</pre>
4.	<p>Mouse over</p> <pre>.navTitleOver .titleHandle</pre>	<pre>.navTitleOver .titleHandle {   background-color: #428eff; }</pre>
5.	<pre>.tree</pre>	<pre>.tree {   background-color: #d6dff7; }</pre>
6.	<pre>.name</pre>	<pre>.name a:link, .name a:visited, .name a:active {   color: #215dc6; }  .name a:hover {   color: #428eff; }</pre>
7.	<pre>.nodeActive .name</pre>	<pre>.nodeActive .name {   background-color: #3878bf; }  .nodeActive .name a:link, .nodeActive .name a:hover, .nodeActive .name a:visited, .nodeActive .name a:active {   color: white; }</pre>

N	Selector	CSS code sample
8.	#userInfo	#userInfo { color: #555555; }
9.	#contexthelp	#contexthelp { color: #555555; border-top: 1px solid #A7B8EB; }
10.	body	body { background-image: url(../images/powered_by.gif); background-position: left bottom; background-repeat: no-repeat; }

## Main Area

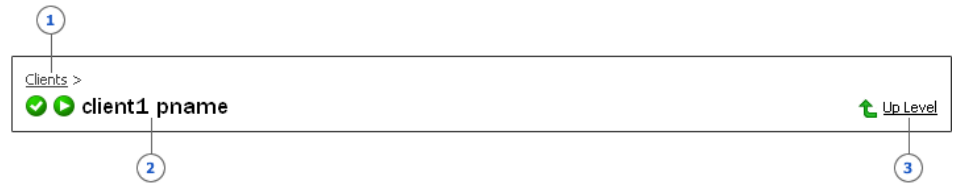
The main area consists of two smaller areas:



1. *screen title* - the title of the currently displayed screen
2. *screen content* - the currently available (visible) set of operations, input forms, lists, etc.

Following is the description of sub-areas that compose the main area and of their elements in detail.

## Screen Title

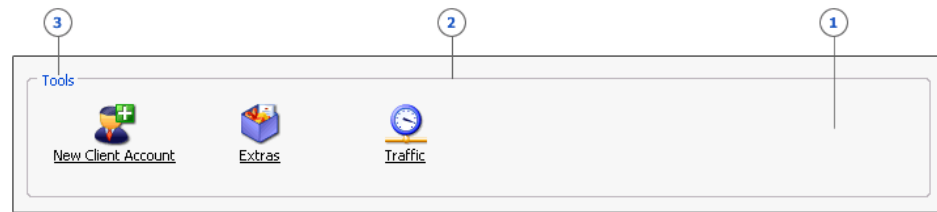


1. *path bar*
2. *title*
3. *'up level' link*

**Table 2.4. Screen title properties**

N	Selector	CSS code sample
1.	.pathbar	<pre>.pathbar {   background: #ffffff;   color: #444444; }  .pathbar a:link, .pathbar a:visited, .pathbar a:hover {   color: #444444; }</pre>
2.	.screenTitle	<pre>.screenTitle {   background: #ffffff; }  .screenTitle td {   font-size: 18px;   font-family: "Franklin Gothic Medium",   Verdana, Arial, sans-serif;   color: #000000; }</pre>
3.	.uplevel .commonButton span  icon can be changed using #bid-up-level in buttons.css	<pre>.uplevel .commonButton span {   text-decoration: underline; }  in buttons.css:  #bid-up-level span {   background-image:   url(../images/btn_uplevel_bg.gif); }</pre>

## General Screen Content



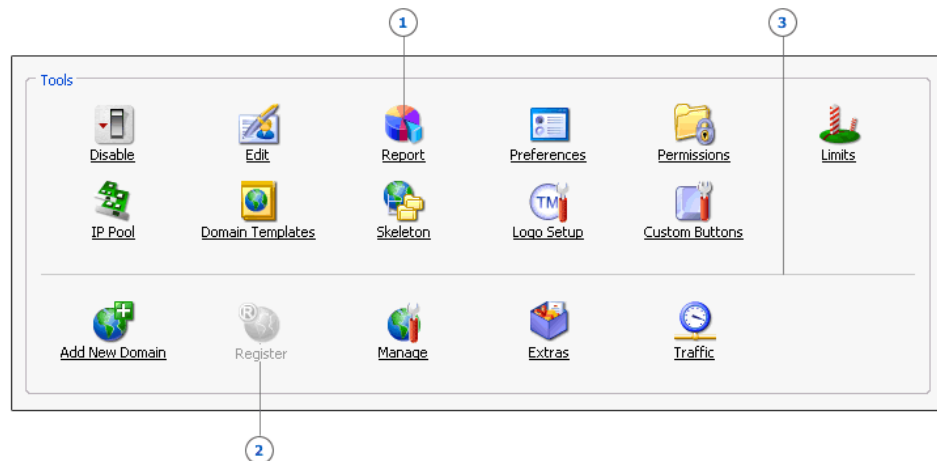
1. *screen content background*
2. *fieldset for grouping ui elements*
3. *fieldset title*

**Table 2.5. General screen content properties**

N	Selector	CSS code sample
1.	body	<pre>body {   background: #F9F8F8; }</pre>
2.	fieldset	<pre>fieldset { }</pre> <p>Presently not available, using default value.</p>
3.	legend	<pre>legend {   color: #0046D5; }</pre>

## Tools

The set of operations at the current screen:



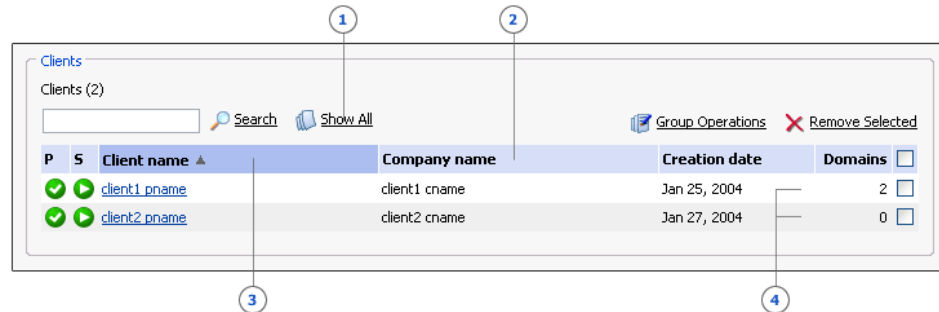
1. *tool*
2. *tool (disabled)*
3. *separator*

**Table 2.6. Tools properties**

N	Selector	CSS code sample
1.	<p><code>.toolsArea .commonButton</code></p> <p>icons are customized through id's (e.g. <code>#bid-report</code>) in <code>buttons.css</code></p>	<pre>.toolsArea .commonButton {   text-decoration: underline; }  in buttons.css:  #bid-report {   background-image: url(../../images/btn_report_bg.gif); }</pre>
2.	<p><code>.toolsArea span.commonButton</code></p> <p>icons are customized through id's (e.g. <code>#bid-report</code>) in <code>buttons.css</code></p>	<pre>.toolsArea span.commonButton {   color: #999999;   text-decoration: none; }  in buttons.css:  #bid-register-disabled {   background-image: url(../../images/btn_register-disabled_bg.gif); }</pre>
1.	<p><code>hr</code></p>	<pre>hr {   color: #cccccc;   background-color: #cccccc;   height: 1px; }</pre>

## Lists

The list of objects:



1. *operations on lists*
2. *table header*
3. *table header (list sorted by selected parameter)*
4. *table's rows*

**Table 2.7. Lists properties**

N	Selector	CSS code sample
1.	<p><code>.buttons .commonButton span</code></p> <p>icons are customized through id's (e.g. <code>#bid-report</code>) in <code>buttons.css</code></p>	<pre>.buttons .commonButton span {   text-decoration: underline; }  in buttons.css:  #bid-show-all span {   background-image:   url(../images/btn_show-all_bg.gif); }</pre>

N	Selector	CSS code sample
2.	th	<pre> th {   text-align: left;   background: #D6DFF7;   border-right: 1px solid #ffffff;   border-bottom: 1px solid #ffffff; }  th a:link, th a:visited {   color: #000000;   text-decoration: none; }  th a:hover {   text-decoration: underline; } </pre>
3.	.sort	<pre> .sort {   background-color: #ABBEEF; } </pre>
4.	.oddrowbg - for odd rows .evenrowbg - for even rows	<pre> .evenrowbg {   background-color: #F0F0F0; }  .oddrowbg {   background-color: #ffffff; } </pre>

## Dialog Forms

1. *parameter name*
2. *'required' indicator*
3. *footnote*
4. *button*

**Table 2.8. Dialog forms properties**

N	Selector	CSS code sample
1.	.name	.name { font-weight: bold; color: #555555; }
2.	.required	.required { color: #cc0000; }
3.	.footnote	.footnote { color: #666666; }

N	Selector	CSS code sample
4.	<p>.commonButton</p> <p>.buttons .commonButton span</p> <p>icons are customized through id's (e.g. #bid-report) in buttons.css</p>	<pre>.commonButton button {   font-family: Tahoma, Verdana, Arial,   Helvetica, sans-serif;   font-size: 11px;   color: #000000;   background-color: transparent;   background-image:   url(../images/btn_bg.gif);   border: 0 solid white;   background-repeat: no-repeat; }  in buttons.css:  #bid-ok button {   background-image:   url(../images/btn_ok_bg.gif);   padding-left: 8px; }</pre>

## Tabs

Properties of certain system objects can be grouped by means of tabs:

The screenshot shows a web interface titled "Server-wide mail preferences" with a breadcrumb "Server >". Below the title are three tabs: "Preferences", "Black List", and "White List". The "Preferences" tab is selected and highlighted. The main content area contains several sections of settings:

- Maximum letter size and relay mode parameters:**
  - Maximum letter size:  Kbytes
  - Relaying:
    - open
    - closed
    - authorization is required:
      - POP3 lock time  Min
      - SMTP
  - Check the passwords for mailboxes in the dictionary:
- Enable MAPS spam protection:**
- MAPS zones:
- Names for POP3/IMAP mail accounts:**
  - Use of short (webmaster) and full (webmaster@domain.com) POP3/IMAP mail account names is allowed
  - Only use of full POP3/IMAP mail accounts names is allowed

At the bottom right, there are "OK" and "Cancel" buttons. A legend at the bottom left indicates "\* Required fields".



1. *tab item*
2. *active tab*
3. *last tab*
4. *tabs panel*

Properties of tabs are defined in the *tabs.css* file.

**Table 2.9. General properties**

UI element	Selector	CSS code sample
1. Tab item	#tabs a, #tabs li	<pre>#tabs a { display: block; background:url("../icons/tabs/right.gif") no-repeat right top; padding:5px 9px 4px 4px; vertical-align: baseline; text-decoration: none; color: #000000; }  #tabs li { float:left; background:url("../icons/tabs/left.gif") no-repeat left top; margin:0; padding:0 0 0 2px; border-bottom: solid 1px #776655; }</pre>
2. Active tab	#tabs #current	<pre>#tabs #current { border-width: 0; }  #tabs #current { background-image:url("../icons/tabs/left_on.gif"); margin-left: -3px; }  #tabs #current a { background-image:url("../icons/tabs/right_on.gif"); padding:3px 9px 7px 6px; vertical-align: baseline; }</pre>
3. Last tab	#tabs last	<pre>#tabs .last a { background-image:url("../icons/tabs/right_last.gif"); }</pre>

UI element	Selector	CSS code sample
4. Tabs panel	#screenTabs, #tabs, #tabs ul	<pre>#screenTabs { float:left; width: 100%; min-height: 1px; height: 1px; background:#ffffff url("../icons/tabs/bg.gif") repeat-x bottom; }  td &gt; #screenTabs { height: auto; }  #tabs { float:left; width:600px; line-height:normal; white-space: nowrap; background:#ffffff url("../icons/tabs/bg.gif") repeat-x bottom; }  #tabs ul { margin:0; padding: 0px 10px 10px 10px; list-style:none; }</pre>

## Help

The Help pages properties that can be customized:

**Table 2.10. Help properties**

UI Element	Selector	CSS code sample
Header level 1	h1	<pre>h1 {   font-size: 16px; }</pre>
Header level 2	h2	<pre>h2 {   font-size: 14px; }</pre>

## Skin Description

Once you are done with preparing CSS and image files, you can create a description file for it. You can choose to edit the file you copied from a standard Plesk skin, or create a new info.xml file.

Following is the source code that you can use for your description file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PLESKSKIN SYSTEM "pleskskin.dtd">
<PLESKSKIN>
  <INFO>
    <PLESKVERSION>7.1.x</PLESKVERSION>
    <VERSION>1.0.0</VERSION>
    <DATE>2004-08-05</DATE>
    <NAME>
      <LOCALESTRING language="en">Your skin
name</LOCALESTRING>
    </NAME>
    <DESCRIPTION>
      <LOCALESTRING language="en">Your skin
description here</LOCALESTRING>
    </DESCRIPTION>
    <AUTHOR>Your name here</AUTHOR>
  </INFO>
  <SCREENSHOTS>
    <SCREENSHOT>
      <NAME>
        <LOCALESTRING language="en"> Screen title,
e.g. Server Administration page </LOCALESTRING>
      </NAME>
      <DESCRIPTION>
        <LOCALESTRING language="en">This is how the
```

```

control panel's server management
section looks like with my skin</LOCALESTRING>
  </DESCRIPTION>
  <SRC>screenshots/screenshot1.gif</SRC>
<THUMB>screenshots/screenshot1_thumb.gif</THUMB>
  </SCREENSHOT>
</SCREENSHOTS>
</PLESKSKIN>

```

Be sure to place your screenshots and thumbnails to the *screenshots/* directory of the skin. You can include as many screenshots as you wish.

## Note

Inside each NAME or DESCRIPTION element you can have several entries in different languages. This can be useful, for instance, when the Administrator's control panel language is set to German - the corresponding entries in German will be displayed, if supplied.

To include an entry in German, use the <LOCALESTRING> tag with attribute **language="de"**:

```

<DESCRIPTION>
  <LOCALESTRING language="en">Your control panel
skin description here</LOCALESTRING>
  <LOCALESTRING language="de">Dies ist die
Beschreibung Ihres Control Panel Skins</LOCALESTRING>
</DESCRIPTION>

```

For other languages, please, use the respective two-character language codes.

## Preparing a Skin Package for Uploading to the Control Panel

Once the skin contents are prepared, you need to create a skin package in order to be able to install your skin into the control panel. It is recommended that you use your favorite archiver software to pack all the skin files and directories in a zip or tar.gz archive, and then simply upload the created archive file to the control panel's skins repository. If desired, you can create a skin rpm package; to do this, follow the instructions provided below.

## Creating a Temporary Build Directory

The first thing you need to do is to create a temporary build directory. It can be located anywhere on your hard disk. Let it be `/tmp/skin_builds/` for example:

```
# mkdir /tmp/skin_build
```

Next, inside the temporary build directory create the complete path to where the

skin will be located when installed in Plesk.

## Creating the RPM Spec File

The RPM spec (specification) file contains data required for building the RPM package. Below is a sample spec file that could be used for the skin considered in our example. In this sample file:

- Name: the skin package name,
- Version: is the version number of the skin,
- Release: is the release number of the skin,
- License: license type can be GPL, Freeware, Commercial, or other common type.

```
# name of your skin to be called in CP
# quote it by the "shell" rules if it contains spaces or
special characters
%define skinname 'My Skin'

# directory where to place your skin
# you may use any sequence of alphanumeric characters and
underscores
%define skindir my_skin

# path where files of your skin is located
%define source /tmp/my_skin

Name: MySkin
Version: 0.0.1
Release: 1
License: BSD
Group: Applications/Internet
Summary: Example spec for Plesk
# END of customizable part
# you can leave the following unchanged in most cases
Buildroot: /var/tmp/build-%{name}-%{version}
Provides: plesk-skin
Requires: psa >= 7.1

%define pleskdir /usr/local/psa/admin
%define pleskskins %{pleskdir}/htdocs/skins

%install
rm -rf $RPM_BUILD_ROOT%{pleskskins}/%{skindir}
mkdir -p $RPM_BUILD_ROOT%{pleskskins}
umask 022
cp -r %{source} $RPM_BUILD_ROOT%{pleskskins}/%{skindir}

%clean
test $RPM_BUILD_ROOT != /
rm -rf $RPM_BUILD_ROOT%{pleskskins}/%{skindir}

%description
This is a sample skin for Plesk 7.1.
```

```
%pre
%{pleskdir}/sbin/skinmng --test-install-directory
--installdir=%{skindir}

%post
%{pleskdir}/sbin/skinmng --register
--installdir=%{skindir} --name=%{skinname}

%preun
%{pleskdir}/sbin/skinmng --remove --installdir=%{skindir}
--leave-files

%files
%defattr(-,root,root)
%{pleskskins}/%{skindir}/
```

To complete preparing the spec file for the custom skin from our example (my\_skin), the above items need to be replaced with proper values.

## Building the RPM Package

Once you have the temporary build directory with the skin files all set and the spec file complete you can proceed to building the RPM package for your skin.

### NOTE

You must be logged in as root to build the RPM package.

Execute the following command:

```
# rpmbuild -bb SKIN.spec
```

Here SKIN.spec should be replaced with the name of your custom skin spec file described in the previous section.

Once the process of building the RPM package is complete you will find the package (in our example it will be `MySkin-0.0.1-1.noarch.rpm`) in the following directory: `/usr/src/redhat/RPMS/noarch`.

---

## Chapter 3. Installing a Skin

To install a skin to the control panel, you need to upload a skin package to the control panel's skins repository. To do this, follow these steps:

1. Login to Plesk control panel as Admin.
2. Select the Server shortcut in the navigation pane.
3. Click the Skins icon in the Control Panel group.
4. Click Add New Skin.
5. Specify the skin package file location and click OK.

Once the skin package is in the repository, you can apply it to the control panel. To do this, at the Server Administration page click Preferences, select your skin, and click OK.

### Installing the Skin RPM Package Using the Command Line

Upload the custom skin RPM package to Plesk server. Execute the following command (we consider the MySkin-0.0.1-1.noarch.rpm from our example):

```
# rpm -Uvh MySkin-0.0.1-1.noarch.rpm
```

The custom is now installed to the control panel skins repository.

To apply it to the control panel, at the Server Administration page click Preferences, select your skin, and click OK.