



Abiquo Branding

Version 2.0 20120316

Date	Revision Information
20110801	Version 1.8 by Xavier Thevenot
20111014	Reorganized and revised to include the Degrafa library
20120316	Version 2.0

Copyright Notice

Copyright ©2012, Abiquo Holdings S.L. All rights are reserved. This documentation may not be reproduced in whole or in part in any manner or form (including photocopying or storing it in any medium by electronic means and whether or not transiently or incidentally to some other use of this documentation) other than in accordance with any applicable license agreement or with the prior written consent of Abiquo Holdings S.L. Any copies of this documentation must incorporate this notice.

Abiquo and the Abiquo logo are trademarks of Abiquo, Inc. Other trademarks used may be owned by third parties.

Contents

1. Branding	4
1.1. Introduction to Themes	4
1.1.1. The Base Theme	4
1.1.2. Theme Structure	5
1.1.3. The Theme Descriptor	5
1.1.4. The SWF Files	6
1.2. How to Create a New Theme	6
1.2.1. CSS Modification	6
1.2.2. Image Modification	6
1.2.3. Distribution	7
1.3. Enabling Branding	7
1.3.1. themes.properties File	7
1.3.2. Database Permissions	8
1.3.3. Custom Enterprise Logos	9
1.3.4. Custom Base Elements	10
Preloader Image	10
Preloader Progress Bar Color	10
Login Logo	10
1.4. Associating a Theme with an Enterprise	11
2. Tutorial on Creating a New Theme	13
2.1. Introduction to Theme Modification	13
2.2. Prerequisites	13
2.2.1. Base Theme from the Theme Kit	13
2.2.2. Flex Builder or Flex SDK	13
2.2.3. Flex 3 CSS Knowledge	13
2.2.4. Degrafa Library for Modifications to base.css	14
2.3. The Client Flex Theme Kit	14
2.4. Replacing Assets	14
2.4.1. Copy the sample theme	14
2.4.2. Replace the Users icon	15
2.4.3. Change the background color	15
2.5. Editing the CSS Files	16
2.6. Compiling the Theme Files	16
2.6.1. Download and Install the Free Flex SDK Compiler	16
2.6.2. Download and Install the Degrafa Library for base.css	16
2.6.3. Compile the affected CSS files	17
2.7. Deploying Your Theme	18
3. Updates to the GUI Between Versions	19
3.1. Version 1.8 to Version 2.0	19
3.1.1. assets	19
3.1.2. application.css	21
3.1.3. main.css	21
3.1.4. virtualimage.css	22
3.1.5. user.css	22
3.1.6. abicloud_premium.css	23

1. Branding

The Branding feature allows you to customize the entire platform with new themes (color, assets, logo, etc.) and to individually manage the theme of each enterprise. This document is intended to give a general overview of the branding feature and guidelines on how to enable it.

1.1. Introduction to Themes

Since version 1.7.0, all elements related to the application's style have been externalized to improve the scalability of the design. The new theme concept allows you to create custom designs for enterprises. A theme is a group of files including a base theme, a theme descriptor, image files and Shockwave Flash (SWF) files.

1.1.1. The Base Theme

The base theme is the first theme loaded when the application starts and it is common to all enterprises. It determines the basic look of the application. If you need to change some of the elements declared in the base theme, place the new elements in the base theme folder. The base theme has the following structure:



- **base.swf**: basic styles for loading the application
- **logo.png**: the logo that will display in the header of the login form [*note: ignore the logo.swf file in the base directory*]
- **theme.lst**: the base theme descriptor
- **CustomPreloaderLogo.png**: the logo displayed during the loading process
- **VirtualImage_defaultIcon.png**: the default Virtual Image icon

1.1.2. Theme Structure

A theme has the following structure:

```
theme
  |_theme.lst
  |_logo.png
  |_abicloud_premium.swf
  |_application.swf
  |_ConfigurationComponentsHeartBeat.swf
  |_ConfigurationComponentsManagement.swf
  |_infrastructure.swf
  |_main.swf
  |_metering.swf
  |_networking.swf
  |_user.swf
  |_virtualappliance.swf
  |_virtualdatacenter.swf
  |_virtualimage.swf
```

- **theme.lst**: the theme descriptor
- **logo.png**: the logo that will display in the header of the application
- **abicloud_premium.swf**: styles for premium features
- **application.swf**: styles for the application framework
- **ConfigurationComponentsHeartBeat.swf**: styles for the heartbeat window
- **ConfigurationComponentsManagement.swf**: styles for the configuration form
- **infrastructure.swf**: styles for infrastructure content
- **main.swf**: styles for the main application, not the framework
- **metering.swf**: styles for metering and statistics content
- **networking.swf**: styles for networking content
- **user.swf**: styles for user data content
- **virtualappliance.swf**: styles for virtual appliances content
- **virtualdatacenter.swf**: styles for virtual datacenters content
- **virtualimage.swf**: styles for virtual images content

1.1.3. The Theme Descriptor

The most important part of the theme is the descriptor because this file tells the application what to load in order to change its appearance.

The theme is fully loaded when all components in the descriptor list are loaded.



Default base theme folder location

/opt/abiquo/tomcat/webapps/client-premium/themes/base

```
theme.lst

//The name of *.swf files to be loaded
infrastructure
virtualimage
application
main
networking
virtualappliance
virtualdatacenter
user
metering
ConfigurationComponentsManagement
ConfigurationComponentsHeartBeat
abicloud_premium
```

1.1.4. The SWF Files

The SWF (Shockwave Flash) files contain all data related to styles and images within the application. A SWF file is generated from a CSS (Cascading Stylesheet) file, which includes CSS declarations and embedded images. Please see the [CSS Modification](#) section for further information.

1.2. How to Create a New Theme

For each new theme you wish to create, you will make a copy of the Abiquo template file and modify it to suit your requirements. Follow the guidelines in the [Tutorial on Creating a New Theme](#).

1.2.1. CSS Modification

In the CSS files, you can change the appearance of a single element or group of elements by modifying the attributes of these elements.

- Do not add/change/delete any CSS name declarations because the application uses the exact names declared in the stylesheets.
- You can change properties (color, font-weight, etc.) but you must use correct values or the application may become unusable.

After modifying the CSS files, convert them into *.swf files, for example, using the [Adobe Flex Framework](#). If you modify base.css, also use the [Degrafa Library for Flex](#).

1.2.2. Image Modification

The template contains all assets used within the application, so you can replace each image with a custom one.

New assets must match the ones to be replaced with:

- the same file names
- the same file type (JPG, PNG, SWF, etc.)
- the same dimensions.

Also, do not add or delete assets, otherwise the application may not work properly.

1.2.3. Distribution

When you have made all your changes, create a new folder with your theme's name and add [all files](#) required for the theme to work.

The folder created must be placed into the themes folder of the application, and the [themes.properties](#) file must updated.



Default theme folder location

/opt/abiquo/tomcat/webapps/client-premium/themes

1.3. Enabling Branding



Branding changes are overwritten on updates

All branding changes will be overwritten each time the **client-premium.war** package is deployed, i.e. each time you update Abiquo.

Please make a backup of the entire **client-premium** folder within the webapps folder before redeploying the **client-premium.war** package, then restore the themes folder into the new version's **client-premium** folder and the themes.properties file in the config folder to continue using this feature.

1.3.1. themes.properties File

The first step in enabling branding is to create a file named **themes.properties** in the client config folder.

```
[root@localhost ~]# cd
/opt/abiquo/tomcat/webapps/client-premium/config/
[root@localhost ~]# touch themes.properties
```



Default client config folder location

/opt/abiquo/tomcat/webapps/client-premium/config

After you have created the `themes.properties` file, edit it as follows:

themes.properties

```
client.themes=abicloudDefault,defaultBlue,defaultGreen,defaultGrey,defaultRed
client.themes.names=Abiquo Default,Blue,Green,Grey,Red
client.ui.enabled=true
client.default.theme=abicloudDefault
client.default.logo=themes/abicloudDefault/logo.png
```

- **client.themes:** defines the available themes (folder name), comma separated, without any space in between
- **client.themes.names:** defines the visible name for each of the above themes
- **client.ui.enabled:** true/false defines if an admin user can change the theme for an enterprise
- **client.default.theme:** the default theme loaded (folder's name)
- **client.default.logo:** the default logo loaded

Be careful when editing the **themes.properties** not to add " " (spaces) after the "=" and "," characters, as this may cause problems in the application.

1.3.2. Database Permissions

To enable branding, you must modify a database property. To do this, place the **kinton-premium-themes.sql** script from the branding Theme Kit into the config folder and execute it.

Example 1: Recommended Abiquo configuration with MySQL password set up.

```
[root@localhost /]# cd
/opt/abiquo/tomcat/webapps/client-premium/config/
[root@localhost config]# mysql -u {dbuser} -p {dbpassword}
<kinton-premium-themes.sql
```

Example 2: Default Abiquo configuration with no MySQL password.


```
[root@localhost ~]# cd
/opt/abiquo/tomcat/webapps/client-premium/config/
[root@localhost config]# mysql -u root
<kinton-premium-themes.sql
```

1.3.3. Custom Enterprise Logos

In addition to the custom enterprise branding described above, this feature allows you to define a specific logo for each enterprise independently of the selected theme. The application tries to load the logo specified in the Edit Enterprise dialog in PNG or JPEG format first. If the logo fails to load, the application will load the one specified in the System Properties/General dialog.

To add a logo for an enterprise, copy it to the enterprise logo folder and give it an appropriate name (the asset's name must match the name specified in the Edit/Create Enterprise dialog)

Retrieve the `enterprise_id` from the Abiquo database using the following commands:

```
[root@localhost config]# mysql kinton
mysql> select idEnterprise, name from enterprise;
```

Result:

```
+-----+-----+
| idEnterprise | name      |
+-----+-----+
|             | 1 | Abiquo  |
|             | 2 | Company x|
|             | 3 | Company y|
+-----+-----+
3 rows in set (0.00 sec)
```

Create folders 'logos' and the relevant 'idEnterprise' folders where applicable. Upload a logo file (PNG or JPEG) to the newly created folder. Note that the image will not be re-sized, default Abiquo logo is a PNG file which has a dimensions of w129 x h69 and a DPI of 299 with a transparent background.

**Logo folder for each enterprise**

Folder structure for example for 'Company x' would be;

```
/opt/abiquo/tomcat/webapps/client-premium/themes/logos/2/
```

1.3.4. Custom Base Elements

As mentioned above, the application loads a **base** theme containing the graphic elements (preloader image, login logo, progress bar color, background image, etc.). Some elements are easily modifiable without modifying CSS files.

Preloader Image

The application loads a specific image from the following path:

**Preloader image location**

```
/opt/abiquo/tomcat/webapps/client-premium/themes/base/CustomPreloaderLogo.png
```

Replace this image with a custom one, using the same name and format. If you do not use the correct name and format, the application will not function properly.

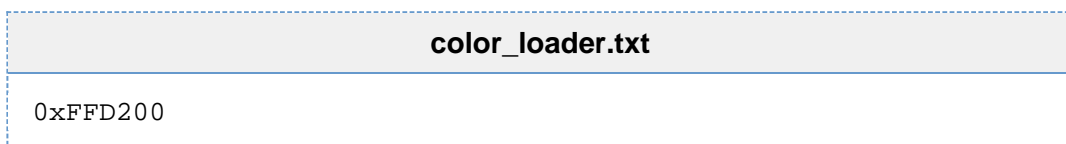
Preloader Progress Bar Color

The preloader progress bar color can be modified by adding the **color_loader.txt** into the base folder.

**Preloader progress bar color location**

```
/opt/abiquo/tomcat/webapps/client-premium/themes/base/color_loader.txt
```

The file contains a hexadecimal value referring to a specific color.



Change this value to a custom one and the application will automatically use it. Use correct hexadecimal color notation (e.g. 0XRRGGBB) or the progress bar will become invisible.

Login Logo

Changing the login logo is similar to changing the preloader image. Add the **logo.png** file in

the following path, using the same name and format to ensure the application functions properly:



Login logo location

/opt/abiquo/tomcat/webapps/client-premium/themes/base/logo.png

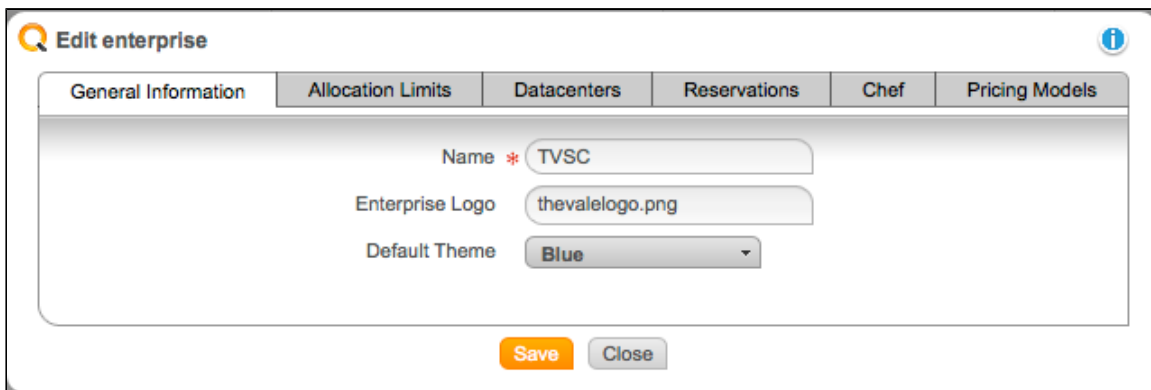
1.4. Associating a Theme with an Enterprise

After you have enabled branding, you should assign the **Modify enterprise theme** privilege to a role in the GUI. This privilege will only be visible on the roles screen after branding is enabled. By default it is not assigned to any role.

Privilege:
Modify enterprise theme

When you create or edit an enterprise, you will see these fields in the first tab:

- **Name:** the enterprise name (required)
- **Enterprise Logo:** the enterprise logo, which can have following values:
 - empty: the displayed logo will be the default one specified in **Configuration -> System Properties -> General -> Default Enterprise logo**
 - valid value: the valid value is the name of the image located in **themes/logos/*enterprise_id/***
 - invalid value: if the path or name specified does not exist, the default image will be displayed
 - logo file should be a .png, .jpg or other web compatible image file.
 - resolution should be 72 ppi and the image size should be 130 x 70 pixels.
- **Default Theme:** the enterprise theme



When you have finished creating the new theme, log in as a user associated with this enterprise and you should see the new theme.

the vale

Home Virtual Datacenters Apps library Users Events Reports Documentation Support

TVSC Admin
TVSC
Logout

ALL

Users

More actions

Name	Username	Email	Role	Activated
test1	test1	test1@test.com	USER (with Virtual Datacenters restriction)	Yes
TVSC Admin	tvadmin	Tv.Admin@abiquo.com	ENTERPRISE_ADMIN	Yes
TVSC User	tvuser	Tv.User@abiquo.com	USER	Yes

Page 1 of 1

2. Tutorial on Creating a New Theme

2.1. Introduction to Theme Modification

This tutorial will show you how to customize the Abiquo GUI by configuring the theme components:

- CSS (Cascading Stylesheet) files
- assets (image files)
- the theme.lst file

The CSS files and the image files are compiled into SWF (Shockwave Flash) files, and the theme.lst file is a descriptor that tells the system which files to use.

You will download a theme kit, which includes a sample theme for this tutorial. You can edit the themes on your local system, by replacing asset files and editing CSS files and then compile the modified files and deploy them to the Abiquo Server.

2.2. Prerequisites

- Base Theme from the ThemeKit
- Flex Builder or Flex SDK
- Flex 3 CSS Knowledge
- Degrafa library for modifications to base.css

2.2.1. Base Theme from the Theme Kit

Contact eesupport@abiquo.com and they will send you the Theme Kit in a zip file. Support can access the file on the internal network from this link ([Base Theme Kit](#)).

2.2.2. Flex Builder or Flex SDK

You must compile the CSS files into SWF files for the application to use. You can do this using the **mxmmlc** utility. To get the mxmmlc command line utility you can install the Flex development environment trial from <http://www.adobe.com/products/flex/> or you can download and install the free SDK from <http://opensource.adobe.com/wiki/display/flexsdk/Download+Flex+3>. For a more detailed explanation of mxmmlc, this is a good starting point <http://www.senocular.com/flash/tutorials/as3withmxmmlc>

2.2.3. Flex 3 CSS Knowledge

You need a basic knowledge of Flex CSS structure to edit the CSS files. CSS for flex does not have the usual web CSS structure, so please see these links for further information:

- CSS Advanced Selectors:
<http://opensource.adobe.com/wiki/display/flexsdk/CSS+Advanced+Selectors>
- Complete -unofficial- list of CSS Properties:

http://www.loscavio.com/downloads/blog/flex3_css_list/flex3_css_list.htm

2.2.4. Degrafa Library for Modifications to base.css

If you wish to modify the base.css file, you must compile it with the Degrafa Library. Download the Degrafa library for flex from http://degrafa.googlecode.com/files/Degrafa_Beta3.1_Flex3.zip.

2.3. The Client Flex Theme Kit

The theme-source.zip file can be requested from eesupport@abiquo.com. Support will send you the zip file downloaded on the internal network from this link ([Base Theme Kit](#)). The zip file contains these folders:

```
assets\  
base-theme\  
css\  
database\  
tutorial\sample_assets  
tutorial\sample_compiled_blue_theme  
tutorial\sample_no_compiled_theme
```

assets: all the original application assets, with source files (psd).

base-theme: this folder contains all source files related to the base theme.

css: all the application's CSS files.

database: the SQL script to update the database to allow the new branding feature to be used ([how to run the database script](#)).

tutorial\sample_assets: all assets used in this tutorial.

tutorial\sample_compiled_blue_theme: a sample of a final new theme package, including all necessary compiled files.

tutorial\sample_no_compiled_theme: this folder contains the structure of a theme, with all related CSS files and assets. It will be used as a base for new themes.

2.4. Replacing Assets

The assets to replace with the new ones are provided in the theme-source in the assets folder and they are organized by the section in which they are used in the application. The SampleTheme folder of the ThemeKit has the same structure but with the final version of each asset.

2.4.1. Copy the sample theme

Start by making a copy of the tutorial/sample_no_compiled_theme folder and renaming it "myTheme".

2.4.2. Replace the Users icon

Change one of the top icons, for example, the Users icon, which looks like this:

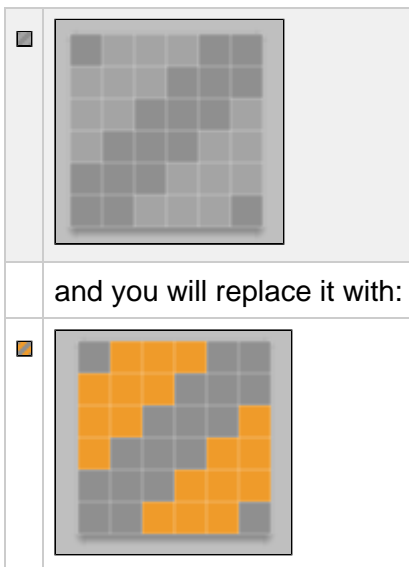


The icon file you are going to replace is located in `theme-source/assets/application/main/header/usersbutton/` and it is called "up_icon.png". You should keep that name for now and just replace the file with your new one.

If you want to change the file name or anything else, you must [edit the CSS file](#). The statement for this icon is located in the file **main.css** and the selector is called **MainHeaderUsersButton**. In this selector you will find a set of CSS attributes for the different icon states (over, down, up) - this is where you can change the name of the file to match the real name.

2.4.3. Change the background color

Replace the background image following the above technique. The background image is currently



So for this step the file you need to replace is `ThemeKit/assets/application/postlogin_background_image.png`

After you make any changes to the theme files, you must compile them following the steps in [Compiling the CSS Files](#) below.

2.5. Editing the CSS Files

In the CSS files you can change the appearance of a single element or group of elements by specifying which elements will have attributes that are different from the defaults.

When you are modifying the CSS files:

- Do not add/edit/delete any CSS name declarations because the application is using the exact list given in the files.
- You can change properties (color, font-weight, etc.), but you must use appropriate values or the application may become unusable.
- After you modify the CSS files, compile them into *.swf files by doing the steps below.
- If you modify the base.css file, remember to follow the steps to use the Degrafa Library

To avoid general style problems, it is best not to edit the Flex basic component styles. If you do edit these styles, you could change, for instance, the appearance of the data grids, which are found throughout the application. The structure sizes (width or height) of data grids, combo boxes, and so on, cannot be modified.

2.6. Compiling the Theme Files

After you modify the theme files (CSS files or assets), compile the affected CSS files into .swf files. The assets are included in the compilation process.

2.6.1. Download and Install the Free Flex SDK Compiler

1. Open this url: <http://opensource.adobe.com/wiki/display/flexsdk/Download+Flex+3>
2. Select the latest 3.X SDK because Abiquo is compiled with this SDK version (tested with version Release Builds 3.5a, size 26MB)
3. Unzip into a folder (e.g. C:\Program Files on Windows)
4. Add SDK path to environment variable. E.g. -> system variables -> PATH = + C:\Program Files (x86)\flex_sdk_3.5\bin on Windows.

2.6.2. Download and Install the Degrafa Library for base.css



Do these steps if you have modified base.css.

1. Download the Degrafa library for Flex from http://degrafa.googlecode.com/files/Degrafa_Beta3.1_Flex3.zip
2. Extract the Degrafa.swc library file in your current working directory (where the base.css file is located). The file name will also include the version number, e.g. Degrafa_Beta3.1_Flex3.zip.
3. After you extract the file, you will see a Degrafa.swc folder with a name that also contains the version number. This is also a compressed archive and you should extract it too.

4. Rename the extracted folder to **degrafa.swc**. Remember that this folder should be located in your working directory; see image.

```
[root@localhost src]# ls -ls
total 148
 4 drwxr-xr-x 6 root root  4096 Feb  3  2011 assets
 8 -rw-r--r-- 1 root root  8017 Oct 10 15:02 base.css
116 -rw-r--r-- 1 root root 111773 Oct 14 07:58 base.swf
 4 drwxr-xr-x 5 root root  4096 Oct 14 07:56 degrafa.swc
 4 -rw-r--r-- 1 root root   981 Oct 10 15:02 logo.css
 4 -rw-r--r-- 1 root root  1136 Feb  3  2011 main.mxml
 4 -rw-r--r-- 1 root root  2373 Feb  3  2011 RepeatedBackground.as
 4 -rw-r--r-- 1 root root    25 Feb  3  2011 theme.lst
[root@localhost src]#
```

2.6.3. Compile the affected CSS files

1. Open a terminal (for example, by running cmd.exe on Windows) and go to your theme folder with source files (*.css), for example, C:\Users\Administrator\Desktop\theme
2. Execute this command for each CSS file you have modified except base.css

```
mxmlc application.css
```

3. If you modified base.css, run this command:

```
mxmlc -external-library-path=degrafa.swc
-runtime-shared-libraries=degrafa.swc/library.swf
base.css
```

```
[root@localhost src]# pwd
/root/dev/1.8.0/abiquo/client/theme/themeBase/src
[root@localhost src]# mxmlc -external-library-path=degrafa.swc -runtime-shared-libraries=degrafa.swc/library.swf base.css
Loading configuration file /opt/flexsdk/frameworks/flex-config.xml
/root/dev/1.8.0/abiquo/client/theme/themeBase/src/base.swf (111773 bytes)
```

If no errors are detected in the CSS file (missing images, for example), the corresponding swf is generated (in this case application.swf or base.swf)

```
C:\Users\Administrator\Desktop\theme>mxmlc application.css
Loading configuration file C:\Program Files (x86)\flex_sdk_3.5\frameworks\flex-c
onfig.xml
C:\Users\Administrator\Desktop\theme\application.swf (145277 bytes)
C:\Users\Administrator\Desktop\theme>_
```

These files can now be deployed to the Abiquo Server.

2.7. Deploying Your Theme

After you have compiled the CSS and you have the modified assets in place, it is time to put everything together.

For instance, if you have modified the file **application.css**, you have to replace the duplicated theme with the newly compiled **application.swf** file. You should compile all the CSS files that you edited and replace their CSS files. You do not need to replace SWF files when the CSS file was unchanged.

Theme Integration

So the application can use the new theme, place the duplicated theme folder into the application's themes folder, and update or create the themes.properties file as follows. Create a file named **themes.properties** in the client config folder.

```
[root@localhost /]# cd
/opt/abiquo/tomcat/webapps/client-premium/config/
[root@localhost themes]# touch themes.properties
```



Default client config folder location

/opt/abiquo/tomcat/webapps/client-premium/config

After you have created the file, edit it as follows:

themes.properties

```
client.themes=abicloudDefault,defaultBlue,defaultGreen,defaultGrey,defaultRed,MyTheme
client.themes.names=Abicloud Default,Blue,Green,Grey,Red,My Enterprise Theme
client.ui.enabled=true
client.default.theme=abicloudDefault
client.default.logo=themes/abicloudDefault/logo.png
```

Where "My Theme" is the folder name of the new theme and "My Enterprise Theme" is the theme name.

The meaning of the properties is as follows:

- **client.themes:** defines the available themes (folder name), comma separated, without any space in between
- **client.themes.names:** defines the visible name for each of the themes
- **client.ui.enabled:** true/false defines if an admin user can change the theme for an enterprise

- **client.default.theme**: the default theme loaded (folder name)
- **client.default.logo**: the default logo loaded



Avoid CSS Problems

Be careful not to add " " (spaces) after the "=" and after "," characters when editing the **themes.properties** file because this may cause problems in the application.

3. Updates to the GUI Between Versions

This document describes updates to the GUI between versions that affect branding. It also includes important notes about upgrading branding.

Each section will include all changes needed and all new/modified assets to include.

3.1. Version 1.8 to Version 2.0

This page describes updates to the GUI between version 1.8.0 and 2.0. After you update branding you should clear your browser cache and check that you can see the branding changes.



Upgrading with Branding Enabled

If you already had branding enabled in version 1.8.0, then you will need to run the database script again when you upgrade to version 2.0. See the Database Permissions section under [Branding](#).

In each theme: client/theme/

- abicloudDefault
- themeBlue
- themeGreen
- themeGrey
- themeRed

Here is the [archive](#) including new assets for version 2.0

3.1.1. assets

Added

Add in `$_THEME_FOLDER/src/assets/application/main/header/pricingbutton/`





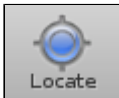
Added

Add in `$_THEME_FOLDER/src/assets/virtualimage/`



Added

Add in `$_THEME_FOLDER/src/assets/infrastructure/standardLocatorLightButton/`



Added

Add in `$_THEME_FOLDER/src/assets/virtualimage/`



Added

Add in `$_THEME_FOLDER/src/assets/application/icons/`



Added

Add in `$_THEME_FOLDER/src/assets/infrastructure/standardLocatorLightButton/`



Modified

This info icon has changed. Modify in `$_THEME_FOLDERsrc/assets/application/icons/info.png` by following one.



Then compile the **main.css** file to into it into swf theme file

3.1.2. application.css

Added

Add these lines at the end of this file: `$_THEME_FOLDER/src/application.css`

```
.CustomProgressBar {  
  bar-color: #FE9608;  
  border-color: #666666;  
  track-height: 14;  
  color: #505050;  
}
```

3.1.3. main.css

Added

Add these lines at the end of this file: `$_THEME_FOLDER/src/main.css`

```

.MainHeaderPricingButton
{
padding-left: 5;
padding-right: 5;
color: #212121;
text-rollover-color: #212121;
text-selected-color: #000000;
font-size: 10;
font-weight: bold;

up-icon: Embed("/assets/application/main/header/pricingbutton/up_icon.png");
over-icon: Embed("/assets/application/main/header/pricingbutton/over_icon.png");
down-icon: Embed("/assets/application/main/header/pricingbutton/down_icon.png");
selected-up-icon: Embed("/assets/application/main/header/pricingbutton/up_icon.png");
selected-over-icon: Embed("/assets/application/main/header/pricingbutton/over_icon.png");
selected-down-icon: Embed("/assets/application/main/header/pricingbutton/down_icon.png");

up-skin: Embed("/assets/nullPixel.png");
over-skin: Embed("/assets/nullPixel.png");
down-skin: Embed("/assets/application/main/header/over_skin.png", scaleGridTop="7",
scaleGridBottom="45", scaleGridLeft="7", scaleGridRight="10");

selected-up-skin: Embed("/assets/application/main/header/selected_up_skin.png", scaleGridTop="7",
scaleGridBottom="45", scaleGridLeft="7", scaleGridRight="10");
selected-over-skin: Embed("/assets/application/main/header/selected_up_skin.png", scaleGridTop="7",
scaleGridBottom="45", scaleGridLeft="7", scaleGridRight="10");
selected-down-skin: Embed("/assets/application/main/header/selected_up_skin.png", scaleGridTop="7",
scaleGridBottom="45", scaleGridLeft="7", scaleGridRight="10");
}

```

3.1.4. virtualimage.css

Added

Add these lines at the end of this file: `$_THEME_FOLDER/src/virtualimage.css`

```

.VirtualImageMiniatureBaseChefImage
{
source:Embed("/assets/virtualimage/chefEnabled.png");
}

```

3.1.5. user.css

Added

Add these lines at the end of this file: `$_THEME_FOLDER/src/user.css`

```
.MissingAllowedDatacenter {
  color: "#ff0000";
  font-weight: bold;
}

.AllowedDatacenter {
  color: "#505050";
  font-weight: normal;
}
```

3.1.6. abicloud_premium.css

Added

Add these lines at the end of this file: `$_THEME_FOLDER/src/abicloud_premium.css`

```
.StandardLocatorLightButton {
  up-skin: Embed(
    "/assets/infrastructure/standardLocatorLightButton/up_skin.png" );
  over-skin: Embed(
    "/assets/infrastructure/standardLocatorLightButton/over_skin.png"
  );
  down-skin: Embed(
    "/assets/infrastructure/standardLocatorLightButton/down_skin.png"
  );
  disabled-skin: Embed(
    "/assets/infrastructure/standardLocatorLightButton/disabled_skin.png"
  );
}

.physicalMachineSTATEHALTEDFORSAVE { /**custom property**/
  source: Embed('/assets/application/icons/halted.png');
}

.UCSLocateFlash { /**custom property**/
  source: Embed(
    "/assets/infrastructure/standardLocatorLightButton/over_flash.png" );
}
```