

Anaconda Progress Visual Style

Alain Reguera Delgado

March 30, 2010

Abstract

This article describes the visual style of Anaconda Progress. Anaconda Progress screen is shown after configuration and while packages are installed. Anaconda Progress screen has a header, and slide images which alternate with release notes texts. This article describes Anaconda Progress slide images and release notes only. The header component of Anaconda Progress is described in the document “Anaconda Header Visual Style”.

Anaconda is the name of the install program used by CentOS. It is python-based with some custom modules written in C. The Anaconda installer works on a wide variety of Linux-based computing architectures (ia32, Itanium, Alpha, S/390, PowerPC), and is designed to make it easy to add platforms.

Copyright © 2010 The CentOS Project. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Contents

1	Introduction	3
2	Installation	4
2.1	Subversion	4

2.2	Inkscape	4
2.3	ImageMagick	5
2.4	Netpbm	5
2.5	Syslinux	5
2.6	GNU Image Manipulation Program	6
2.7	GNU Core Utilities	6
2.8	L ^A T _E X	6
3	Configuration	7
3.1	Firewall	7
3.2	Subversion Behind Squid	7
	3.2.1 Subversion Client Configuration	7
	3.2.2 Squid Server Configuration	7
3.3	Working Copy	8
3.4	User Identification	9
	3.4.1 Account Registration	9
	3.4.2 Account Privileges	9
3.5	Shell Environment	9
	3.5.1 Relative Paths	9
	3.5.2 Absolute Paths	10
	3.5.3 Environment Variable Definition	10
	3.5.4 Environment Variable Initialization	10
4	Framework	10
4.1	Identity	11
	4.1.1 Design Templates	11
	4.1.2 Design Templates Export Id	13
	4.1.3 Design Models	13
	4.1.4 Image Files	13
	4.1.5 Image Files Rendering	18
4.2	Translations	18
	4.2.1 Common Translations	18
	4.2.2 Specific Translations	18
	4.2.3 Translation Path	19
	4.2.4 Translation Markers	21
	4.2.5 Translation Rendering	21
	4.2.6 Translation Rendering Script	23
	4.2.7 Translation Template Directory	24

4.3	Manuals	24
4.3.1	Writing Style	24
4.3.2	File Structure	24
4.3.3	Exporting To PDF	26
4.4	Scripts	26
4.4.1	Invocation Scripts	27
4.4.2	Configuration Scripts	27
4.4.3	Function Scripts	27
5	Rebranding	29
5.1	Images	29
5.1.1	redhat-logos	29
5.2	Messages Locale	29
5.3	Release Notes	31
5.3.1	centos-release-notes	31
6	GNU Free Documentation License	32
6.1	Applicability And Definitions	32
6.2	Verbatim Copying	34
6.3	Copying In Quantity	35
6.4	Modifications	35
6.5	Combining Documents	38
6.6	Collections Of Documents	38
6.7	Aggregation With Independent Works	39
6.8	Translation	39
6.9	Termination	40
6.10	Future Revisions Of This License	40
6.11	How to use this License for your documents	40

1 Introduction

Anaconda Progress Visual Style is organized inside CentOS Artwork Repository.

2 Installation

This section describes the tools you need to have installed in your CentOS workstation in order to interact with your working copy of CentOS Artwork Repository.

2.1 Subversion

Subversion is used to interact with CentOS Artwork Repository.

Subversion is a version control system, which allows you to keep old versions of files and directories (usually source code), keep a log of who, when, and why changes occurred, etc., like CVS, RCS or SCCS.¹

To install Subversion client tools in your workstation you can use the following command:

```
yum install subversion
```

2.2 Inkscape

Inkscape is used to design and render images inside CentOS Artwork Repository.

Inkscape is a GUI editor for Scalable Vector Graphics (SVG) format drawing files, with capabilities similar to Adobe Illustrator, CorelDraw, Visio, etc. Inkscape features include versatile shapes, bezier paths, freehand drawing, multiline text, text on path, alpha blending, arbitrary affine transforms, gradient and pattern fills, node editing, SVG-to-PNG export, grouping, layers, live clones, and more.

Note that Inkscape is not inside CentOS Distribution, so you need to configure a third party repository like RPMForge or EPEL to install Inkscape. Installation of a third party repositories inside CentOS Distribution is described in the following URL:

<http://wiki.centos.org/AdditionalResources/Repositories>

¹More documentation about Subversion and its tools, including detailed usage explanations of the svn, svnadmin, svnserve and svnlook programs, historical background, philosophical approaches and reasonings, etc., can be found at <http://svnbook.red-bean.com/>.

Once you have configured the third party repository you can install Inkscape using the following command:

```
yum install inkscape
```

2.3 ImageMagick

ImageMagick is used by scripts inside CentOS Artwork Repository.

ImageMagick is a free software suite for the creation, modification and display of bitmap images. It can read, convert and write images in a large variety of formats. Images can be cropped, colors can be changed, various effects can be applied, images can be rotated and combined, and text, lines, polygons, ellipses and Bzier curves can be added to images and stretched and rotated.

To install ImageMagick in your workstation you can run the following command:

```
yum install ImageMagick
```

2.4 Netpbm

Netpbm is used by scripts inside CentOS Artwork Repository.

Netpbm is a toolkit for manipulation of graphic images, including conversion of images between a variety of different formats. There are over 300 separate tools in the package including converters for about 100 graphics formats.

To install Netpbm in your workstation you can run the following command:

```
yum install netpbm{-progs}
```

2.5 Syslinux

The package `syslinux` provides the programs `ppmtolss16` and `lss16toppm` which are used to produce Anaconda Prompt images. The `ppmtolss16` Perl program also includes the file format specification.

Syslinux is a suite of bootloaders, currently supporting DOS FAT filesystems, Linux ext2/ext3 filesystems (EXTLINUX), PXE network boots (PXELINUX), or ISO 9660 CD-ROMs (ISOLINUX). It also includes a tool, MEMDISK, which loads legacy operating systems from these media.

To install Syslinux in your workstation you can run the following command:

```
yum install syslinux
```

2.6 GNU Image Manipulation Program

GNU Image Manipulation Program (GIMP) is used to manipulate images inside CentOS Artwork Repository.

To install GIMP in your workstation you can run the following command:

```
yum install gimp
```

2.7 GNU Core Utilities

The GNU core utilities are a set of tools commonly used in shell scripts.

To install the GNU core utilities in your workstation you can run the following command:

```
yum install core-utils
```

2.8 L^AT_EX

L^AT_EX is used to create manuals inside CentOS Artwork Repository.

L^AT_EX is a document preparation system implemented as a macro package for Donald E. Knuth's T_EX typesetting program. The L^AT_EX command typesets a file of text using the T_EX program and the LaTeX Macro package for T_EX. To be more specific, it processes an input file containing the text of a document with interspersed commands that describe how the text should be formatted.

To install L^AT_EX in your workstation you can run the following command:

```
yum install tetex-{latex,fonts,doc,xdiv,dvips}
```

3 Configuration

This section describes the configuration steps you need to do in your CentOS workstation in order to interact with your working copy of CentOS Artwork Repository.

3.1 Firewall

The CentOS Artwork Repository lives on the following URL:

```
https://projects.centos.org/svn/artwork/
```

To reach this location you need to have Internet access and be sure no rule in your firewall is denying this site. Note that the URL uses the SSL protocol (port 443).

3.2 Subversion Behind Squid

Sometimes it is convenient to proxy Subversion client's requests through a proxy-cache server like Squid. In cases like this, the Squid proxy server is in the middle between you and CentOS Artwork Repository. If you want to proxy Subversion client's requests through Squid proxy-cache server, you need to configure your Subversion client and your Squid proxy server to do so.

3.2.1 Subversion Client Configuration

Subversion client needs to be configured to send requests to your Squid proxy-cache server. This configuration takes place in the file:

```
~/.subversion/servers
```

3.2.2 Squid Server Configuration

Squid proxy-cache server needs to be configured to accept the extension methods REPORT MERGE MKACTIVITY CHECKOUT MKCOL. This configuration takes place in the file:

```
/etc/squid/squid.conf
```

specifically in the configuration text described below:

```
# TAG: extension_methods
#     Squid only knows about standardized HTTP request methods.
#     You can add up to 20 additional "extension" methods here.
#
#Default:
# none
extension_methods REPORT MERGE MKACTION CHECKOUT MKCOL
```

3.3 Working Copy

A Subversion working copy is an ordinary directory tree on your local system, containing a collection of files (i.e. Translations, Designs, Manuals, and Scripts). You can edit these files however you wish. Your working copy is your own private work area: Subversion will never incorporate other people's changes, nor make your own changes available to others, until you explicitly tell it to do so. You can even have multiple working copies of the same project.²

After you've made some changes to the files in your working copy and verified that they work properly, Subversion provides you with commands to "publish" your changes to the other people working with you on your project (by writing to the repository). If other people publish their own changes, Subversion provides you with commands to merge those changes into your working directory (by reading from the repository).

To download your working copy of CentOS Artwork Repository in the location `~/Desktop/artwork`, run the following command:

```
svn co https://projects.centos.org/svn/artwork ~/Desktop/
```

The previous command will download lots of files into your workstation. This process may take some time. When finish you are ready to start exploring and improving available works.

²Even this is basically correct, doing so when using CentOS Artowrk Repository can bring some confusion when executing scripts. Presently, only one absolute path can be defined as absolute path for scripts' execution. You can have as many working copies of CentOS Artwork Repository as you want but scripts will be executed from just one working copy absolute path—the one you defined in the variable `CENTOS_ARTWORK_WC`. For more information about this, see section 3.5.

3.4 User Identification

At this point you probably have made some changes inside your working copy and wish to publish them. To publish your changes you need to have a registered account with commit privilege in CentOS Artwork Repository.

If you are new in CentOS Artwork Repository it is possible that you can't commit your changes. That is because new registered accounts haven't commit privilege set by default. In order for your registered account to have commit privilege inside CentOS Artwork Repository you need to request it. See section 3.4.2.

3.4.1 Account Registration

To register a user account inside CentOS Artwork Repository, you need to go to the following URL:

```
https://projects.centos.org/trac/artwork/
```

3.4.2 Account Privileges

To have commit privileges in CentOS Artwork Repository it is needed that you show your interest first, preferably with something useful like a new or improved design, translation, manual, or script. As convention, people working on CentOS Artwork Repository share ideas in the mailing list `centos-devel@centos.org`. If you are interested in joining us go there and express yourself.

3.5 Shell Environment

Inside CentOS Artwork Repository we try to automate tasks as much as possible using shell scripts. In order to execute shell scripts we need to provide their path, relative or absolute.

3.5.1 Relative Paths

Using relative paths makes difficult to call functions from different levels inside the directory structure of your working copy of CentOS Artwork Repository.

3.5.2 Absolute Paths

Using absolute paths lets us use functions from different levels inside the directory structure of your working copy of CentOS Artwork Repository but creates an inconvenience. It forces us to use an unchangeable absolute path to store the working copy of CentOS Artwork Repository.

3.5.3 Environment Variable Definition

To avoid forcing a predefined absolute path to store our working copy of CentOS Artwork Repository, we decided to define the **CentOS_ARTWORK_WC** environment variable. The value of this environment variable contains the absolute path used to store our working copy of CentOS Artwork Repository in the workstation. This way we can download our working copy of CentOS Artwork Repository wherever we want and be sure that scripts inside it execute correctly.

3.5.4 Environment Variable Initialization

As CentOS_ARTWORK_WC environment variable defines the absolute path used to execute scripts, it isn't possible to initialize it inside scripts themselves. To initialize CentOS_ARTWORK_WC environment variable we use the personal initialization file (`~/.bash_profile`), executed for login shells.

For example, if you downloaded your working copy of CentOS Artwork Repository in the absolute path `~/Desktop/artwork`, then you need to add the following lines to your personal initialization file:

```
CentOS_ARTWORK_WC=~/Desktop/artwork
export CentOS_ARTWORK_WC
```

For changes to take effect you need to logout and do login again.

4 Framework

Anaconda Progress framework is inside your working copy of CentOS Artwork Repository and organized in the following sections:

4.1 Identity

trunk/Identity/Themes/\$THEME/Distro/Anaconda/Progress/

Here is where Anaconda Progress design templates and image rendering take place. Anaconda Progress Identity file structure is illustrated in Figure 1 and described in the following subsections.

4.1.1 Design Templates

trunk/Identity/Themes/\$THEME/Distro/Anaconda/Progress/template/

Here is where Anaconda Progress design templates are stored. Design templates in this location are used to control the look and feel of “first slide” and “language-specific slides” images used by Anaconda’s package installation process.

The first slide (see Figure 5) is the one used to open the package installation process. The first slide image is controlled by `first.svg`, and `first-lowres.svg` design templates. If the screen resolution is less than 800 x 600 pixels, then the ‘-lowres’ design is used.

The language-specific slides images (see Figure 6, and Figure 7) start to rotate after the first slide. The language-specific slides images are controlled by `list.svg` and `paragraph.svg` design templates.

In CentOS Distribution, language-specific slides images are used to resume current distribution’s release notes and features. As graphic designer, you don’t need to care very much about translating language-specific slides images, this is a job for translator guys (see section 4.2). As graphic designer, most of your attention is focused on how the slide images look like.

To verify the final look and feel of your first slide and language-specific slides images, you use the `render.sh` identity script (see section 4.1.5) which takes care of getting translations and apply them to your slide designs using translation markers (see section 4.2.4).

During the package installation process, language-specific slides images are loaded based on Anaconda’s installation language. When Anaconda’s installation language is different to English, Anaconda looks for language-specific slides images that match the current Anaconda’s installation language and uses them, if they exist. Otherwise, if there is no language-specific slides images available for the current Anaconda’s installation language, Ana-

```
trunk/Identity/Themes/$THEME/Distro/Anaconda/Progress/
|-- img
|   |-- 3
|       |-- bn_IN
|           |-- 01-welcome.png
|           |-- 02-donate.png
|           |-- 03-yum.png
|           |-- ... (more bn_IN images here)
|       |-- cs
|           |-- 01-welcome.png
|           |-- 02-donate.png
|           |-- 03-yum.png
|           |-- ... (more cs images here)
|       |-- ... (more languages here)
|       |-- first-lowres.png
|       |-- first.png
|       |-- ... (more languages here)
|       |-- progress_first-lowres.png
|       |-- progress_first.png
|       |-- ... (more languages here)
|   |-- 4
|   |-- 5
|   |-- 6
|   '-- ... (more major releases here)
|-- render.sh
'-- template
    |-- first-lowres.svg
    |-- first.svg
    |-- list.svg
    '-- paragraph.svg
```

Figure 1: Anaconda Progress identity framework.

conda uses the English language-specific slides images.

In contrast to language-specific slides images, the first slide images has no translation and they are used as they are, no matter what the current Anaconda’s installation language be.

4.1.2 Design Templates Export Id

Every object inside design templates have an “export id” property. By default the export id is a combination of names and numbers granting its uniqueness. Inside CentOS Artwork Repository, design templates are rendered automatically using the `render.sh` identity script. The `render.sh` identity script looks for the design area containing the `CENTOSARTWORK` uppercase word as its export id and exports it as bitmap.

If you are designing templates for CentOS Artwork Repository, and you are using the `render.sh` identity script to produce images, you need to use the `CENTOSARTWORK` uppercase word as export id in your design templates. This way, you define the design area you want to export when using the `render.sh` identity script.

4.1.3 Design Models

`trunk/Identity/Models/Distro/Anaconda/Progress/`

Design models are representative images used to illustrate key components inside a specific design. Design models are frequently used to make documentation clearer. Anaconda Progress design models are described in Figure 3, Figure 4, Figure 5, Figure 6, and Figure 7. File structure is organized in its own framework, illustrated in Figure 2.

When designing models, try to make them language independent so they can be reused in differet language documents. For example, you can use letters or numbers to identify areas in the model and later use the figure’s caption to describe the meaning of those letters and numbers, respectively.

4.1.4 Image Files

`trunk/Identity/Themes/$THEME/Distro/Anaconda/Progress/img/`

Here is where Anaconda Progress final images are stored. Final images are rendered using the `render.sh` identity script (see section 4.1.5).

```
trunk/Identity/Models/Distro/Anaconda/Progress/  
|-- img  
|   |-- build-0.pdf  
|   |-- build-0.png  
|   |-- build-1.pdf  
|   |-- build-1.png  
|   '-- ... (more images here)  
|-- render.sh  
'-- template  
    |-- build-0.svg  
    |-- build-1.svg  
    '-- ... (more templates here)
```

Figure 2: Anaconda Progress design models framework.



Figure 3: Anaconda Progress slide images. A = “Header”, B = “Slide rotation”, C = “Action/Navigation”.

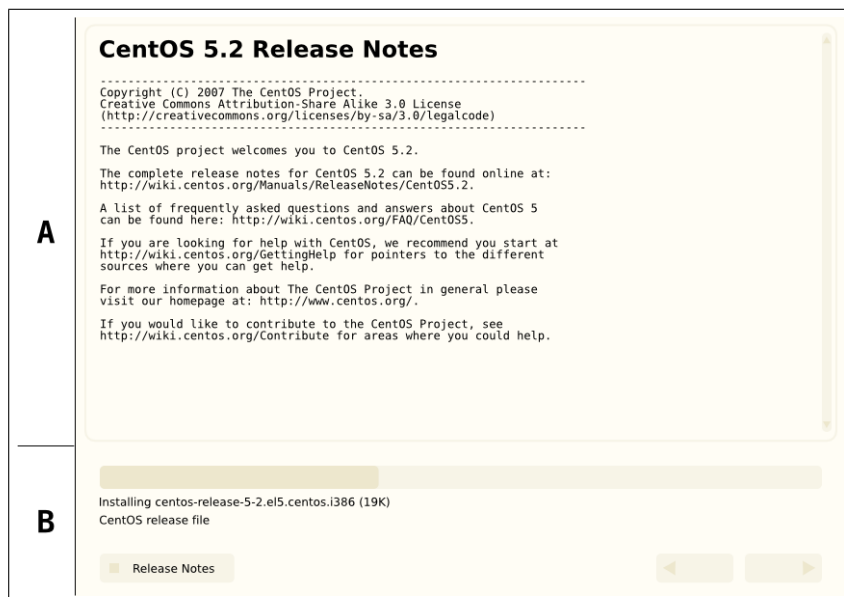


Figure 4: Anaconda Progress release notes. A = “Release notes”, B = “Action/Navigation”.

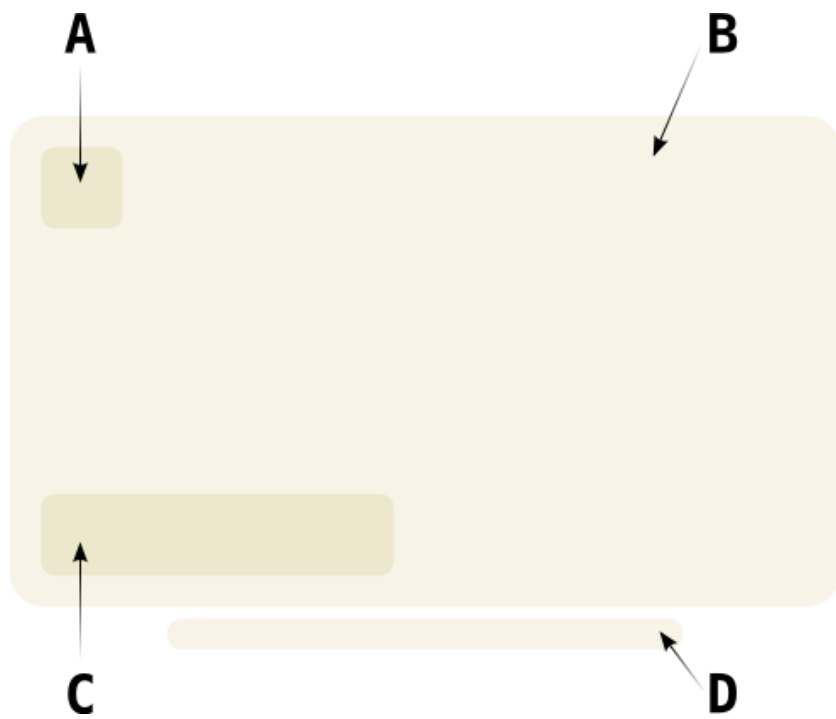


Figure 5: Anaconda Progress first slide template. A = “The CentOS Symbol”, B = “The CentOS Default Artistic Motif”, C = “The CentOS Release Brand”, D = “The CentOS Copyright”.



Figure 6: Anaconda Progress list template.

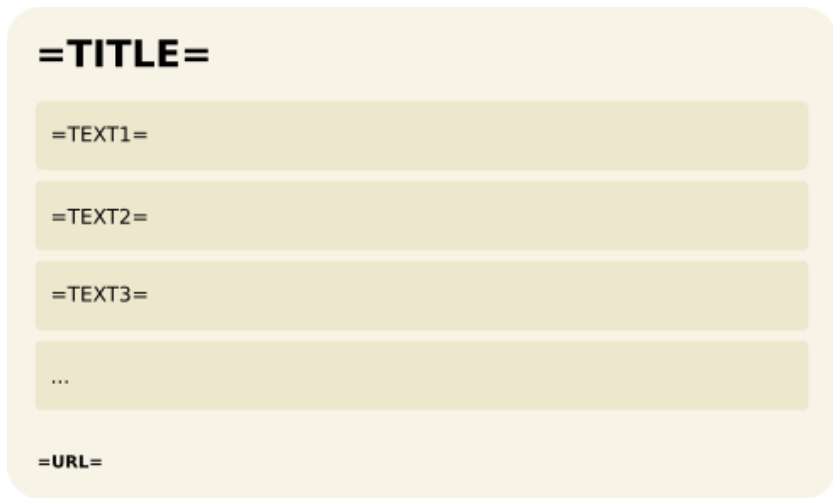


Figure 7: Anaconda Progress paragraph template.

In the rendering process, final images are organized automatically by `render.sh` identity script. To organize final images, the `render.sh` identity script takes the related translation path as reference (see section 4.2).

Final images are what you use to rebrand (see section 5).

4.1.5 Image Files Rendering

The `render.sh` identity script applies translation files to design templates to produce translated images automatically. Use the `render.sh` identity script whenever you need to create translated images. The `render.sh` identity script has the following syntax:

```
./render.sh 'REGEX'
```

The REGEX argument is optional. It is used to reduce the amount of images you want to render. It is a posix-egrep regular expression pattern, applied against the translation path.

4.2 Translations

trunk/Translations/Identity/Themes/Distro/Anaconda/Progress/

Here is where translators locale images. Image localization is defined inside `.sed` files, also known as translation files. Translation files can be common or specific. The given organization of translation files defines the translation path.

4.2.1 Common Translations

Common translation files contain common localization or no localization at all for their related images. They are in the root directory of the translation path. Common translation files create common images for all major releases of CentOS Distribution.

4.2.2 Specific Translations

Specific translation files contain specific localization for their related images. Specific translation files are not in the root directory of the translation path. Specific translation files are inside directories which describe the type of translation they are doing.

4.2.3 Translation Path

Translation path is where we organize common and specific translation files. Translation path is also used as reference to build the path of rendered images inside image directory (see section 4.1.4). Anaconda Progress translation path is illustrated in Figure 8.

When rendering images, if no REGEX argument is provided to `render.sh` identity script, all translation files in the translation path are read and applied one by one to its related design template —defined in the configuration script— to produce a translated image. Images produced using the `render.sh` identity script have the same name of its translation file, but with the `.png` extension instead.

To control the number of images produced by `render.sh` identity script, you need to look into the translation path and provide a regular expression pattern that match the translation path, or paths, related to the image, or images, you want to produce. For example, based on the translation path shown in Figure 8, to produce the CentOS 5 Anaconda Progress Slides, in English language only, you can do:

```
./render.sh '5/(progress|first|en)'
```

To produce CentOS 5 and CentOS 6 Anaconda Progress Slides in English and Spanish languages only, you can do:

```
./render.sh '(5|6)/(progress|first|en|es)'
```

To produce the `01-welcome.png` slide for CentOS 5 and CentOS 4 Anaconda Progress, in English and Spanish languages only, you can do:

```
./render.sh '(5|4)/(en|es)/01-welcome'
```

The regular expression pattern you provide to `render.sh` identity script is applied to the translation path from its very beginning. It is not the same to say `5/es/01-welcome` that `01-welcome`, the first expression match but the last one does not.

When using REGEX you don't need to specify the file extension. It is removed from translation path before applying the REGEX pattern, so it doesn't count here.

Translation path shown in Figure 8 is an incomplet version of the real one. It was cropped in the sake of keeping it in just one page. To make

```

trunk/Translations/Identity/Themes/Distro/Anaconda/Progress/
|-- 3
|   |-- bn_IN
|   |   |-- 01-welcome.sed
|   |   |-- 02-donate.sed
|   |   |-- 03-yum.sed
|   |   |-- ... (more bn_IN translation files)
|   |-- ... (more language directories)
|   |-- first-lowres.sed
|   |-- first.sed
|   |-- ... (more language directories)
|   |-- progress_first-lowres.sed
|   |-- progress_first.sed
|   |-- ... (more language directories)
|-- ... (more release versions)
|-- render.sh
'-- template
    |-- bn_IN
    |   |-- 01-welcome.sed
    |   |-- 02-donate.sed
    |   |-- 03-yum.sed
    |   |-- ... (more bn_IN translation files)
    |-- ... (more language directories)
    |-- first-lowres.sed
    |-- first.sed
    |-- ... (more language directories)
    |-- progress_first-lowres.sed
    |-- progress_first.sed
    |-- ... (more language directories)

```

Figure 8: Anaconda Progress translation framework.

Marker	Description
=VERSION=	Major release number of CentOS Distribution.
=TITLE=	Slide's title.
=DESCRIPTION=	Slide's list description.
=TEXT1-12=	Slide's content.
=URL=	Slide's URL.

Table 1: Anaconda Progress translation markers.

yourself a better idea of the real Anaconda Progress translation path, check the one inside your working copy of CentOS Artwork Repository. That is the one you should use in order to build your REGEX patterns when rendering Anaconda Progress slides.

4.2.4 Translation Markers

Translation markers are used in design templates and translation files as replacement pattern to commit image translation. When `render.sh` identity script renders images, translation files are applied to design templates to get a PNG translated image as result. In order to have the appropriate translation on the PNG image, marker definition in translation files should match markers in design templates.

Translation markers can be whatever text you want, but as convention we've defined those we use inside CentOS Artwork Repository. This definition creates a common point of reference to translators and graphic designers groups. To have a clean definition of translation markers is what makes possible that translators and graphic designers can work together but independently.

In Anaconda Progress, translation files and design templates use the translation markers described in Table 1. A translation file example using translation markers is illustrated in Figure 9.

4.2.5 Translation Rendering

Anaconda Progress translation paths and files —also known as the translation structure— are produced using the `render.sh` translation script, available in the current translation directory.

The `render.sh` translation script combines the content of a translation

```
# Warning: Do not modify this file directly. This file is created
# automatically using render.sh translation script. Any change you
# do in this file will be lost the next time you run render.sh
# translation script. If you want to improve the content of this
# translation file, improve its template file instead and run the
# render.sh translation script later to propagate changes.
# -----
# $Id: 01-welcome.sed 4959 2010-03-18 02:27:24Z al $
# -----

s/=TITLE=/Welcome to CentOS =MAJOR_RELEASE= !/
s/=TEXT1=/Thank you for installing CentOS =MAJOR_RELEASE=./
s/=TEXT2=/CentOS is an enterprise-class Linux Distribution derived\
from sources freely provided to the public by a prominent North\
American Enterprise Linux vendor./
s/=TEXT3=/CentOS conforms fully with the upstream vendors\
redistribution policy and aims to be 100% binary compatible. CentOS\
mainly changes packages to remove upstream vendor branding and\
artwork./

s/=TEXT4=//
s/=TEXT5=//
s/=TEXT6=//
s!=URL=!http://www.centos.org/!

# Release number information.
s!=RELEASE=! =MAJOR_RELEASE=. =MINOR_RELEASE=!g
s!=MINOR_RELEASE=!0!g
s!=MAJOR_RELEASE=!5!g
```

Figure 9: English translation file for CentOS 5 Anaconda Progress ‘welcome’ slide. Files like this are used by `render.sh` identity script to produce translated images.

template directory —holding common information— with the release-specific information you provide as argument when executing the script. As result, the `render.sh` translation script produces the translation structure that `render.sh` identity script needs to create translated images.

4.2.6 Translation Rendering Script

The `render.sh` translation script produces release-specific translation directories. Use the `render.sh` translation script whenever you need to create a new release-specific translation directory based on translation template directory. The `render.sh` translation script has the following syntax:

```
./render.sh RELEASE ...
```

The `RELEASE` argument defines the release number used to create the release-specific translation directory. You can pass many `RELEASE` arguments, separated by one or more spaces, to `render.sh` translation script in a single call. If no `RELEASE` argument is passed to `render.sh` translation script then all release-specific translation directories, available in the current translation directory, are updated using the translation template as reference.

Default behaviour of `render.sh` translation script may be usefull if all your release-specific translation directories always have the same information that translation template does. If this is the case, you only need to maintain the translation template and use the `render.sh` translation script to propagate changes to each release-specific translation directory.

In contrast, if you are using the `render.sh` translation script to create release-specific translation directories that use translation template as base to introduce non-reusable translations, you should take care when executing the `render.sh` translation script. Otherwise, your non-reusable translations may be replaced with those in template.

Inside CentOS Artwork Repository, all translation directories (reusable and non-reusable) are versioned. If you accidentally propagate template content to a non-reusable translation, you have the Subversion's `revert` and `export` commands to undo the mess.

As general rule, when you use the `render.sh` translations script, take care of details and check twice before commit your translation changes up to CentOS Artwork Repository.

4.2.7 Translation Template Directory

The translation template directory is located in the current translation directory and contains common translations for all release-specific translation directories. It is also used as base to build non-reusable translation.

4.3 Manuals

trunk/Manuals/Identity/Themes/Distro/Anaconda/Progress/

Here is where “Anaconda Progress Visual Style” manual is prepared. If you want to help improving Anaconda Progress Visual Style manual this is where you need to go.

4.3.1 Writing Style

When writing manuals for CentOS Artwork Repository, keep in mind the following quote taken from the online “BBC News Styleguide”: —The key to good writing is **simple thoughts simply expressed**. Use short sentences and short words. Anything which is confused, complicated, poorly written or capable of being misunderstood risks losing the listener or viewer, and once you have done that, you might just as well not have come to work—.

If you need to express complicated ideas, try to split them out in smaller and simpler ideas as much as possible. If you consider it appropriate, try to use design models (see section 4.1.3) to illustrate your thoughts.

4.3.2 File Structure

The file structure of Anaconda Progress Visual Style manual is illustrated in Figure 10 and Figure 11. Relevant files in this structure are described below:

Manual.tex is the main \LaTeX document. Here is where you define specific document information like class, title, author, abstract, etc. Here is where you organize the loading order of individual files containing specific sections. The Anaconda Progress Visual Style **Manual.tex** file is specific to Anaconda Progress Visual Style manual and it is not reused by other manuals.

introduction.tex extends the introduction provided in **Manual.tex** abstract. The Anaconda Progress Visual Style **introduction.tex** file is

```
trunk/Manuals/Identity/Themes/Distro/Anaconda/Progress/  
|-- Manual.aux  
|-- Manual.log  
|-- Manual.lot  
|-- Manual.out  
|-- Manual.pdf  
|-- Manual.tex  
|-- Manual.toc  
'-- sections  
    |-- framework.tex  
    |-- introduction.tex  
'-- rebranding.tex
```

Figure 10: Specific documentation files. These files are specific to Anaconda Progress Visual Style manual only.

```
trunk/Manuals/  
|-- Licenses  
|   |-- CentOS-Redistribution.tex  
|   '-- GFDL.tex  
|-- configuration.tex  
'-- installation.tex
```

Figure 11: Common documentation files. These files are reused by all manuals.

specific to Anaconda Progress Visual Style manual and it is not reused by other manuals.

installation.tex provides information about what you need to install in your workstation in order to interact with CentOS Artwork Repository. The **installation.tex** file is common to and reused by all manuals inside CentOS Artwork Repository.

configuration.tex provides information about what you need to configure in your workstation before interact with CentOS Artwork Repository. The **configuration.tex** file is common to and reused by all manuals inside CentOS Artwork Repository.

framework.tex provides information about how to interact with CentOS Artowrk Repository, specifically to the Anaconda Progress area. The Anaconda Progress Visual Style **framework.tex** file is specific to Anaconda Progress Visual Style manual and it is not reused by other manuals.

rebranding.tex provides information about how to rebrand Anaconda Progress using rendered images in CentOS Artowrk Repository. The Anaconda Progress Visual Style **rebranding.tex** file is specific to Anaconda Progress Visual Style manual and it is not reused by other manuals.

4.3.3 Exporting To PDF

To produce Anaconda Progress Visual Style PDF manual, you need to get inside Anaconda Progress Visual Style manual's file structure and execute the command:

```
pdflatex Manual.tex
```

4.4 Scripts

Inside CentOS Artwork Repository, scripts are organized in three groups: “invocation scripts”, “configuration scripts” and “function scripts”. These scripts are mainly used to help us automate and standardize tasks.

4.4.1 Invocation Scripts

Invocation scripts are identified by the name `render.sh`. They are specific to each section and help us to reduce the amount of code used to perform tasks. Instead of copying the same code all around CentOS Artwork Repository, invocation scripts provide a way to call the task code from a common place. Invocation scripts are distributed all around CentOS Artwork Repository, specially under `trunk/Identity/` and `trunk/Translations/` root directories. The specific work of invocation scripts is to call the appropriate configuration script.

4.4.2 Configuration Scripts

`trunk/Scripts/Config/Identity/Themes/Distro/Anaconda/Progress/`

Here is where Anaconda Progress “configuration scripts” are stored. The configuration script is identified by the name `render.conf.sh` and is the first script called when the `render.sh` invocation script is executed by you. In the configuration script you can load function scripts and define what and how translation files apply design templates.

There are different types of configuration scripts. It depends on the task the specified configuration script respond to. For example when you are rendering images, the `render.sh` invocation script that you execute calls a configuration script designed to realize image rendering. Likewise, when you are rendering texts, the `render.sh` invocation script that you execute calls a configuration script designed to realize text rendering.

Inside configuration scripts, the `SVG` variable defines how translations are applied to design templates. To render Anaconda Progress correctly, the `SVG` configuration variable should be defined as illustrated in Figure 12. If you need to add or remove slide images from the list, you can modify the value of `SVG` variable to fit your needs.

4.4.3 Function Scripts

Function scripts are used inside configuration scripts to do specific tasks. Function scripts are common for all configuration scripts and can be reused inside themselves.

```
# Template and Translation matching list. Define which design template
# is applied to which translation file.
SVG="\
first-lowres.svg:\
    first-lowres.sed\
    progress_first-lowres.sed
first.svg:\
    first.sed\
    progress_first.sed
paragraph.svg:\
    01-welcome.sed\
    02-donate.sed\
    03-yum.sed\
    04-repos.sed\
    05-centosplus.sed\
    06-support.sed\
    08-wiki.sed\
    09-virtualization.sed
list.svg:\
    07-docs.sed
"
```

Figure 12: Anaconda Progress configuration layout.

5 Rebranding

To comply with upstream redistribution policy, the CentOS Project removes all upstream brands and artworks from CentOS Distribution. The CentOS Project has its own brand and its own artwork. The CentOS Brand and CentOS Artwork are what the CentOS Project uses in CentOS Distribution.

The action of removing upstream brands and artworks and add CentOS brands and artworks is what we call rebranding.

CentOS Brands and artworks are organized inside CentOS Artwork Repository. The CentOS Artwork Repository is maintain by CentOS Artwork SIG which is formed by CentOS Community People.

When rebranding use original names as much as possible. Do not rename original file names if you don't need to. To rebrand the original file information, update just the file content using the 'cp' command or something similar.

5.1 Images

This section describes relation between SRPM packages and image files you need to modify in order to rebrand Anaconda Progress correctly.

5.1.1 redhat-logos

The `redhat-logos` package contains files created by the CentOS Project to replace the Red Hat “Shadow Man” logo and RPM logo. The Red Hat “Shadow Man” logo, RPM, and the RPM logo are trademarks or registered trademarks of Red Hat, Inc. Anaconda Progress images that need to be rebranded in `redhat-logos` package are illustrated in Figure 13.

Replacements for these files are available in the image directory (see section 4.1.4) of Anaconda Progress Identity (see section 4.1) inside your working copy of CentOS Artwork Repository.

Once you rebrand the image files inside `redhat-logos` SRPM package, you need to rebuild it with the new brand information.

5.2 Messages Locale

Anaconda Progress doesn't have messages locale. Anaconda Progress images are translated as described in section 4.1, and section 4.2.

```
/usr/share/anaconda/pixmaps/  
|-- first-lowres.png  
|-- first.png  
|-- progress_first-lowres.png  
|-- progress_first.png  
|-- rnotes  
|   |-- 01-centos5-welcome.png  
|   |-- 02-centos5-donate.png  
|   |-- 03-centos5-yum.png  
|   |-- 04-centos5-repos.png  
|   |-- 05-centos5-centosplus.png  
|   |-- 06-centos5-support.png  
|   |-- 07-centos5-docs.png  
|   |-- 08-centos5-wiki.png  
|   |-- 09-centos5-virtualization.png  
|   |-- cs  
|   |   |-- 01-centos5-welcome.png  
|   |   |-- 02-centos5-donate.png  
|   |   |-- 03-centos5-yum.png  
|   |   |-- 04-centos5-repos.png  
|   |   |-- 05-centos5-centosplus.png  
|   |   |-- 06-centos5-support.png  
|   |   |-- 07-centos5-docs.png  
|   |   |-- 08-centos5-wiki.png  
|   |   |-- 09-centos5-virtualization.png  
|   |-- ... (more languages here)
```

Figure 13: Anaconda Progress slide images.

5.3 Release Notes

During the installation process Anaconda provides a button labeled “Release Notes” (see Figure 3). When this button is pressed the header and slide areas get hidden and the available space is used to display CentOS release notes (see Figure 4).

Presently, CentOS release notes are managed online and they don’t appear in Anaconda’s release notes screen. A few paragraphs are used instead to describe how CentOS release notes are managed and how they can be accessed.

5.3.1 centos-release-notes

The `centos-release-notes` package contains Anaconda Progress release notes files. Anaconda Progress release notes files are illustrated in Figure 14.

```
/usr/share/doc/centos-release-notes-5.2/  
|-- RELEASE-NOTES-cs  
|-- RELEASE-NOTES-cs.html  
|-- RELEASE-NOTES-de  
|-- RELEASE-NOTES-de.html  
|-- RELEASE-NOTES-en  
|-- RELEASE-NOTES-en.html  
|-- RELEASE-NOTES-es  
|-- RELEASE-NOTES-es.html  
'-- ... (more language-specific release notes)
```

Figure 14: Anaconda Progress release notes files.

Files in Figure 14 have their own framework inside CentOS Artwork Repository. Anaconda Progress release notes are rendered similar to images, using templates and translation files, as well as rendering scripts. For more information about release notes rendering see the “CentOS Release Visual Style” manual.

6 GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

6.1 Applicability And Definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept

the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML,

PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

6.2 Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 6.3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

6.3 Copying In Quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

6.4 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 6.2 and 6.3 above, provided that you release the

Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the

same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

6.5 Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 6.4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6.6 Collections Of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

6.7 Aggregation With Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 6.3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

6.8 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 6.4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 6.4) to Preserve its Title (section 6.1) will typically require changing the actual title.

6.9 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

6.10 Future Revisions Of This License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

6.11 How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) YEAR YOUR NAME.
```

```
Permission is granted to copy, distribute and/or modify this
document under the terms of the GNU Free Documentation License,
Version 1.2 or any later version published by the Free Software
Foundation; with no Invariant Sections, no Front-Cover Texts,
and no Back-Cover Texts. A copy of the license is included in
the section entitled ‘‘GNU Free Documentation License’’.
```


If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts”. line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.